

Diskettenlaufwerk

DOS 2.5: XF551™

Bedienungshandbuch

Inhaltsverzeichnis

Das ATARI Diskettenlaufwerk XF551 und DOS

Das Atari Laufwerk XF551 ist ein effizientes und schnelles Speichermedium, welches perfekt Ihren Atari XE oder XL unterstützt. Der interne Computerspeicher merkt sich alle Eingaben, welche Sie ihm über die Tastatur eingeben. Doch dieser Speicher ist begrenzt, und ohne ein externes Speichermedium, wie das Laufwerk XF551, gehen seine gespeicherten Informationen immer dann verloren, wenn Sie Ihren Computer abschalten.

Das Atari Laufwerk XF551 erlaubt es Ihnen große Informationsmengen zu speichern und zu verwalten. Sie können diese Informationen in sogenannten Ordnern ablegen, welche Sie frei benennen können.

Dies ermöglicht es Ihnen, Ihre gespeicherten Informationen problemlos wiederzufinden oder zu kopieren.

Alles in allem hilft Ihnen somit das Laufwerk Ihre gesammelten Informationen effizient zu verwalten.

Was ist DOS?

Um überhaupt Informationen auf Disketten speichern zu können, benötigen Sie entsprechende Software, welche in der Lage ist eine Kommunikation zwischen Ihrem Computer und dem Diskettenlaufwerk herzustellen.

Das ist der große Augenblick, an dem Sie das Disk Operating System (DOS) kennenlernen !

DOS (sprich: Doss) ist das oben angesprochene Programm, welches Ihrem Computer ermöglicht, mit dem Diskettenlaufwerk in Verbindung zu treten.

DOS selbst ist in mehreren Dateien zusammengefaßt auf der Master-Diskette enthalten.
Bevor Sie mit der Arbeit anfangen, müssen Sie zunächst einmal DOS in Ihren Computer einladen. Manche Programme enthalten bereits eine DOS-Version, sodaß Sie sich das Einladen von DOS bei diesen Programmen ersparen können. Bei Programmen, welche noch keine DOS-Version beinhalten (dies sind insbesondere Steckmodule), müssen Sie dementsprechend zunächst DOS laden, wenn Sie vorhaben, bestimmte Informationen auf Ihrem Laufwerk zu speichern.
Im Grunde genommen benötigen Sie DOS für sehr viele unterschiedliche Vorgänge, wie z.B. Formatieren von Disketten, Speichern von Dateien, oder um sich eine Sicherungskopie zu erstellen.

DOS 2.5 und Ihr ATARI Personal Computer System

Bis auf ein paar kleine Änderungen ist DOS 2.5 kompatibel zu früheren DOS-Versionen, z.B. Atari DOS 2.0.

Dies bedeutet, daß Sie auch mit dem neuen DOS Ihre bereits gespeicherten Informationen weiter bearbeiten können. Sie können DOS 2.5 mit den Laufwerken XF551 und Atari 810 benutzen. Wie auch immer Ihre System-Konfiguration aussieht, auf jeden Fall muß immer ein XF551-Laufwerk enthalten sein.

Mit DOS 2.5 können Sie Ihre Disketten sowohl im einfachen als auch im erweiterten Format formatieren. Im erweiterten Format können Sie ca. 50% mehr Daten auf Ihrer Diskette speichern als bei der DOS-Version 2.0 .
Sie können derart formatierte Disketten jedoch nur mit einem XF551-Laufwerk verwalten, da das Atari Laufwerk 810 dazu nicht mehr in der Lage ist. Auch die auf der DOS 2.0 Systemdiskette enthaltenen Dateien können Sie mit dem Atari 810-Laufwerk nicht bearbeiten. Aus diesem Grund ist es zwingend notwendig mit einem XF551-Laufwerk zu arbeiten, wenn Sie die DOS-Version 2.5 nutzen wollen.

Wenn Ihr Computersystem ein Atari 810-Laufwerk beinhaltet, und Sie öfters damit arbeiten, ist es sicherlich besser, wenn Sie Ihre Disketten im einfachen Format formatieren.

Anmerkung: DOS 2.5 arbeitet sowohl mit dem Laufwerk XF551, als auch mit dem Laufwerk 1050 von Atari. Sollten Sie also über ein Laufwerk 1050 verfügen, so können Sie alle in diesem Handbuch beschriebenen Funktionen auch mit diesem Laufwerk ausführen.

Wenn Ihr System eine XF551 und eine 810 beinhaltet, und Sie unter DOS 2.0 erstellte Dateien mit DOS 2.5 bearbeiten, so beachten Sie bitte den Anhang H. Wenn Sie mit DOS 3.0 arbeiten, beachten Sie bitte den Anhang L.

Bitte benutzen Sie beim Arbeiten mit einem dieser Laufwerke einseitige Disketten mit doppelter Schreibdichte.

DOS 2.5 arbeitet mit allen Steckmodulen, die auf dem Atari Computer lauffähig sind, also auch der AtariWriter-Textverarbeitung und Atari Basic.

Viele der bekannten Modul-Programme sind auf die DOS-Version 2.0 abgestimmt, sind jedoch ebenfalls mit der DOS-Version 2.5 lauffähig.

Das Arbeiten mit dem XF551-Handbuch

Diese Beschreibung wurde sowohl für den Computer-Anfänger, als auch für den erfahrenen Anwender geschrieben.
Sie beinhaltet zwei Teile, welche auf das Laufwerk XF551 und die Fähigkeiten von DOS eingehen. Es beinhaltet ebenso ein Kapitel mit technischen Informationen, welche gerade für den fortgeschrittenen Anwender interessant sein könnten, sowie einige Anhänge.

Besonders der Anhang J, welcher die zahlreichen Befehle beinhaltet, die im Umgang mit DOS anfallen, wird für den Anfänger sehr hilfreich sein.
Wenn Sie mit dieser Beschreibung arbeiten, blättern Sie immer einmal in diesen Anhängen nach - es wird Ihnen die Arbeit wesentlich erleichtern.

Wenn Sie zum ersten Mal ein Disketten-Laufwerk anschließen, so sollten Sie zunächst Kapitel 1 durchlesen, in welchem einfache Instruktionen über das Starten und Benutzen des Laufwerkes stehen.

Das zweite Kapitel "Einführung in DOS 2.5" zeigt Ihnen die am meisten verwendeten Befehle, die unter DOS 2.5 gebraucht werden. Mit kontinuierlich aufbauenden Instruktionen wird Ihnen erklärt, wie Sie DOS in Ihren Hauptspeicher einladen, wie Sie Disketten mit Speicherdateien belegen, wie Sie Ihre Disketten formatieren, wie Sie Ihre gesicherten Daten kopieren, löschen und weiter bearbeiten können.
Auch das für Sie Wichtigste, das Zusammenstellen einer Systemdiskette oder Arbeitskopie, wird Ihnen hier ausführlich und anschaulich erklärt. Dieses sollten Sie auch als ersten Arbeitsschritt tun, bevor Sie mit der eigentlichen Arbeit beginnen.
Um diese Beschreibung durchzuarbeiten, benötigen Sie drei Leerdisketten. Auf der ersten Diskette sollten Sie sich eine Systemdiskette einrichten, die weiteren zwei sollten Sie als Arbeitsdisketten benutzen.

Falls einige der aufgeführten Beispiele auf Atari Basic eingehen, so benötigen Sie ebenfalls noch ein Basic-Modul, falls Sie über einen Atari 400 oder 800 verfügen. Bei den Atari Systemen 130XE, 800XE und 800XL ist dieses Basic bereits im Equipment enthalten, sprich fest im Rechner installiert.

Kapitel 3 zeigt Ihnen die verschiedenen Funktionen von DOS 2.5 in anschaulichen Beispielen.

Kapitel 4 beschreibt die Basic-Kommandos, die Sie unter DOS 2.5 zur Verfügung gestellt bekommen. Dieses Kapitel ist deshalb auch mehr für den fortgeschrittenen Anwender gedacht.

In den Anhängen finden Sie die häufigsten Fehlermeldungen, sowie deren äquivalente Fehlernummer.

Schauen Sie in Anhang C, wenn Sie beim Starten von DOS 2.5 eine Fehlermeldung erhalten haben.

Wenn Sie mehrere Laufwerke angeschlossen haben, schauen Sie bitte im Anhang G nach.

Wenn Sie ein 810-Laufwerk Ihr Eigen nennen, oder Dateien haben welche mit einer älteren DOS-Version erstellt wurden, so finden Sie nähere Auskünfte im Anhang H.

Wenn Sie einen Atari 130XE besitzen, so blättern Sie im Anhang K nach.

Wenn Ihre Dateien mit DOS 3.0 erstellt wurden, ist der Anhang L für Sie interessant.

Kapitel 1: Das ATARI Diskettenlaufwerk XF551

Wenn Sie Ihr Laufwerk nun ausgepackt haben, überzeugen Sie sich bitte davon, daß die Lieferung vollständig ist.

Sie müßten folgende Teile vor sich liegen haben:

1 Atari XF551 Diskettenstation
1 Serielles I/O - Kabel
1 Netzteil
1 Masterdiskette DOS 2.5

Wenn Sie eines dieser Teile vermissen, wenden Sie sich bitte sofort an Ihren System-Fachhändler.

Bitte heben Sie die Verpackung auf, damit Sie bei einer eventuellen Einsendung Ihrer Floppy eine solide Verpackung zur Hand haben.

Anschluß des Diskettenlaufwerkes an Ihren Computer

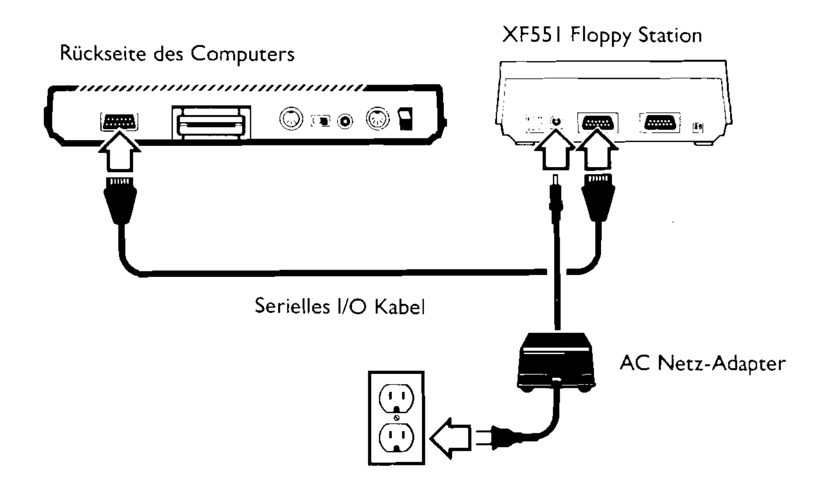
Ihr Computer und Ihr Laufwerk sollten an einem stabilen Platz in der Nähe einer Steckdose stehen.
Setzten Sie Ihr System bitte keiner übermäßigen Hitze, Feuchtigkeit oder Staubbelastung aus.

Befolgen Sie nun bitte die folgenden Arbeitsschritte, um das Laufwerk an den Computer anzuschließen:

1. Versichern Sie sich, daß die Stromzufuhr zu Ihrem Computer und Ihrem Laufwerk ausgeschaltet ist.

2. Stecken Sie ein Ende des seriellen Kabels in die mit Peripherie gekennzeichnete Buchse an der Rückseite Ihres Computers. Das andere Ende des Kabels stecken Sie in eine der mit I/O gekennzeichneten Buchsen an der Rückseite Ihres Laufwerks.

3. Stecken Sie nun die Buchse des Netzteiles in die mit Power gekennzeichnete Buchse an der Rückseite Ihres Laufwerks. Wie Sie sich sicher denken können, gehört das andere Ende in die Steckdose.



4. Nun schalten Sie das Laufwerk an. Es sollte nun die sogenannte Betriebskontroll-Leuchte angehen. Dies bedeutet, daß Ihre Floppy beschäftigt ist. Wenn das Kontrollicht ausgeht, können Sie eine Diskette in das Laufwerk einlegen.

Achtung: Ihr Laufwerk sollte nicht in der Nähe Ihres Fernsehgerätes stehen, da Ihr Fernseher eine starke magnetische Spannung produziert. Dies kann zur Folge haben, daß Ihre auf der Diskette gesicherten Daten zerstört werden.

Ihr Laufwerk ist nun bereit die Systemdiskette DOS 2.5 aufzunehmen. Bitte lesen Sie auf jeden Fall das folgende Kapitel, bevor Sie mit der eigentlichen Arbeit beginnen.

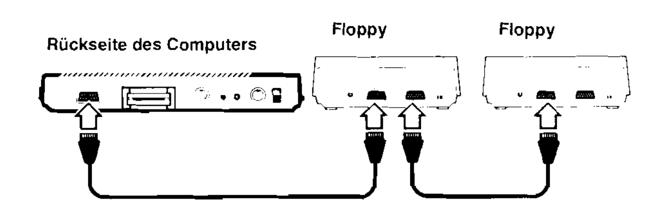
Anschluß von mehreren Diskettenlaufwerken

Sie können maximal vier Diskettenlaufwerke XF551 an Ihren Computer anschließen.
Die Diskettenlaufwerke werden untereinander mit mehreren seriellen I/O-Kabeln verbunden. Wenn Sie mehrere Laufwerke benutzen, trägt dies wesentlich zu einem komfortablen Arbeiten bei.

Bemerkung: Sie können bis zu sechs externe Peripheriegeräte an Ihren Atari XE oder XL anschließen.

Um zwei oder mehr Laufwerke anzuschließen, beachten Sie bitte folgende Instruktionen:

1. Stecken Sie das Ende des seriellen I/O-Kabels in die mit Peripherie gekennzeichnete Buchse an der Rückseite Ihres Computers. Das andere Ende stecken Sie dann wieder in die mit I/O gekennzeichnete Buchse an Ihrem Laufwerk. Nun nehmen Sie ein weiteres I/O-Kabel und stecken das eine Ende in die zweite I/O-Buchse Ihres ersten Laufwerks, das andere Ende kommt dann in eine der I/O-Anschlüsse Ihres zweiten Laufwerks. Bei weiteren Disketten-Laufwerken verfahren Sie in derselben Weise.



2. Nun müssen Sie nur noch die Stromzufuhr zu Ihren Laufwerken sicherstellen. Dies geschieht in der schon in Kapitel 1 beschriebenen Art und Weise.

3. Einrichten der Laufwerk-Kennungen
   Damit Ihr Computer erkennen kann, daß Sie mehrere Laufwerke angeschlossen haben, müssen Sie Ihre Laufwerke nun entsprechend kennzeichnen.

Dies geschieht mit Hilfe zweier kleiner Schalter, welche sich an der Rückseite Ihres Laufwerkes befinden. Es ist nicht notwendig die Laufwerke in einer bestimmten Reihenfolge zu kennzeichen, jedoch muß immer ein Laufwerk als Nummer 1 gekennzeichnet sein. Die verschiedenen Laufwerke dürfen niemals die gleiche Kennzeichnung haben.

Das folgende Bild zeigt Ihnen die Möglichkeiten der Kennzeichnung Ihrer Laufwerke. Ist das untere Feld schwarz unterlegt, so muß sich der Schalter in der unteren Position befinden, ist das obere Feld schwarz unterlegt, muß der Schalter sich in der oberen Position befinden.
Benutzen Sie zum Kennzeichnen Ihrer Laufwerke einen Stift oder einen kleinen Schraubendreher.

Laufwerk	Schalter Einstellung
1	
2	
3	
4	

Schutzmaßnahmen für Ihre Disketten

Disketten sind mit einer sehr empfindlichen magnetischen Schicht überzogen, welche Ihre Daten aufnimmt und von einem Diskettenlaufwerk gelesen werden kann.
Um sicherzustellen, daß Ihre Disketten eine lange Lebensdauer haben, sollten Sie diese pfleglich behandeln.

Jede Diskette ist separat in einem Umschlag verpackt, und wird in einer Papierhülle aufgehoben.

Die meisten Disketten haben eine kleine Einkerbung, welche als Schreibschutz dient. Der Schreibschutz ist aktiviert, wenn Sie diese Einkerbung mit den mitgelieferten selbstklebenden Folien überziehen.

Weitere Informationen finden Sie in Kapitel 2.

Bitte beachten Sie die nachstehend aufgeführten Punkte für die richtige Handhabung und Pflege Ihrer Disketten:

1. Bitte achten Sie beim Einlegen und Herausnehmen Ihrer Disketten darauf, daß die Laufwerks-Kontrolleuchte nicht mehr aufleuchtet.

2. Schalten Sie niemals Ihr Laufwerk an, wenn sich eine Diskette darin befindet. Weiterhin sollten Sie niemals eine Diskette im Laufwerk lassen, wenn das System ausgeschaltet ist.

3. Halten Sie Ihre Disketten fern von Magnetquellen (z.B. Monitore, Fernseher, Elektromotoren und Telefonen)

4. Halten Sie Ihre Disketten fern von direkter Sonneneinstrahlung, Feuchtigkeit und höheren Temperaturen.

5. Vermeiden Sie Berührungen mit der Magnetbeschichtung der Disketten.

6. Beschriften Sie Ihre Disketten-Labels niemals direkt auf der Diskette. Benutzen Sie zum Beschriften keinen Kugelschreiber, sondern lieber einen Filzstift.

7. Vermeiden Sie das Knicken der Disketten. Behandeln Sie Ihre Disketten mit Vorsicht beim Einlegen bzw. Herausnehmen aus dem Laufwerk.

8. Nach Beendigung Ihrer Arbeit sollten Sie Ihre Disketten wieder in die dafür vorgesehenen Papierhüllen zurückstecken, und in einer geschlossenen Diskettenbox aufbewahren.

Beschriften Ihrer Disketten

Die meisten Disketten werden mit einem Satz Labels zusammen ausgeliefert. Versichern Sie sich, daß jede Ihrer Disketten ordnungsgemäß beschriftet ist. Dies ermöglicht Ihnen später ein problemloses Auffinden Ihrer Disketten.

Kapitel 2: Arbeiten mit DOS 2.5

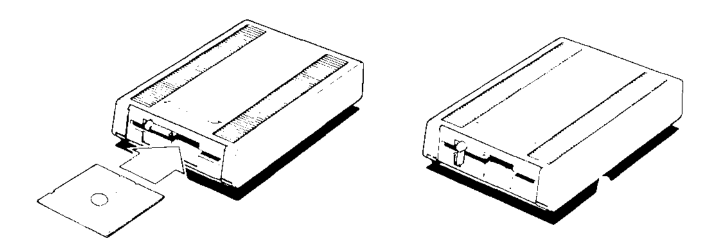
Laden des DOS

Bitte beachten Sie die folgenden Schritte beim Laden von DOS 2.5 in Verbindung mit Atari Basic (wenn Sie das Basic nicht Laden wollen, schauen Sie unter "DOS laden ohne Basic" auf der folgenden Seite nach):

1. Versichern Sie sich das Ihr Computer und Ihr Diskettenlaufwerk ausgeschaltet sind. Wenn Sie einen Atari 400, 800 Computer besitzen, führen Sie das Atari Basic-Steckmodul in Ihren Computer ein. Stellen Sie sicher, daß sich keine Diskette in Ihrem Laufwerk befindet. Wenn Sie einen Atari XE-Spielsystem, 130XE, 800XE oder 800XL mit eingebauten Basic besitzen, versichern Sie sich, daß kein Steckmodul im Computer ist, und keine Diskette sich im Laufwerk befindet.

2. Schalten Sie Ihr Diskettenlaufwerk ein - wenn Sie mehr wie ein Laufwerk besitzen, müssen Sie das DOS immer von Laufwerk 1 laden. Am Laufwerk beginnt nun die Kontrolleuchte zu leuchten.

   Achtung: Niemals eine Diskette in das Laufwerk einlegen, bzw. herausnehmen, wenn die Kontrolleuchte aufleuchtet.



3. Wenn die Kontrolleuchte ausgegangen ist, lösen Sie nun den Hebel an der Vorderseite Ihres Laufwerks. Nehmen Sie die Diskette aus der Papierhülle und führen Sie sie mit der beschrifteten Seite nach oben in Ihr Laufwerk ein, bis Sie ein Klicken vernehmen. Nun bringen Sie den Hebel an der Vorderseite wieder in seine Ursprungsposition zurück (vertikal).

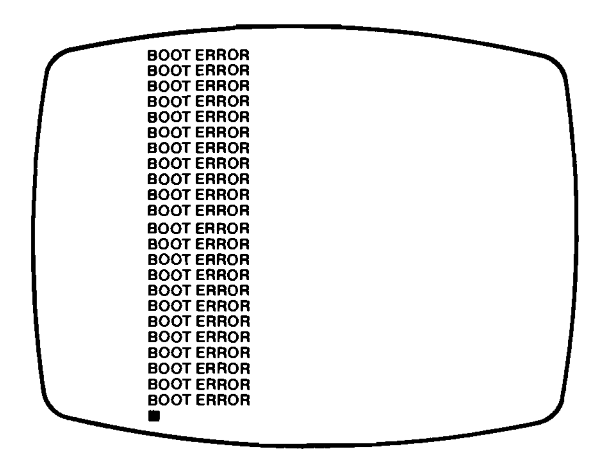
4. Schalten Sie Ihren Computer ein. Die Kontrolleuchte an Ihrem Laufwerk sollte nun wieder aufleuchten, während Ihr Computer das DOS lädt. Wenn Sie die Lautstärke an Ihrem Fernseher hochdrehen, können Sie akustisch das Laden verfolgen.

5. Auf Ihrem Bildschirm müßte nun der Eintrag "Ready" erscheinen. Geben Sie nun DOS ein, und bestätigen Sie mit "Return".

DOS Laden ohne BASIC
Natürlich wollen Sie nicht immer DOS 2.5 zusammen mit Atari Basic laden. Wenn Sie einen Atari 400 oder 800 Computer besitzen, befolgen Sie die oben genannten Schritte ohne das Basic-Steckmodul in den Computer zu stecken. Der oben beschriebene Schritt 5 ist nun natürlich nicht mehr relevant.

Wenn Sie einen Atari XE Spielcomputer, 130XE, 800XE oder 800XL besitzen, ist das Basic bereits eingebaut. Das Basic wird automatisch geladen, sobald Sie Ihren Computer einschalten, selbst dann wenn Sie DOS laden - es sei denn Sie stecken ein Steckmodul in den dafür vorgesehenen Anschluss oder Sie drücken die Taste "OPTION" auf Ihrem Computer während Sie ihn anschalten.

Boot Error
Das Laden eines Programmes in Ihren Computer beim ersten Einschalten wird "Booten" genannt. Wenn dabei in Ihrem System ein Problem auftritt, wird folgende Fehlermeldung auf Ihrem Bildschirm erscheinen:



Beim Starten Ihres Systems können folgende BOOT ERROR auftreten:

1. Auf der eingelegten Diskette ist kein DOS vorhanden.

2. Die Diskette wurde falsch eingelegt.

3. Die Diskette ist zerkratzt, verbogen oder verschmutzt. In diesem Fall sollte man eine andere Diskette benutzen.

4. Die Diskette hat eine vergrößerte Dichte für ein Atari 810 Diskettenlaufwerk.

Weiterhin werden folgende Probleme zu einem BOOT ERROR führen, aber es wird zu keiner Fehlermeldung auf dem Bildschirm kommen:

1. Das Diskettenlaufwerk wurde angeschaltet, nachdem der Computer angeschaltet wurde.

2. Das Diskettenlaufwerk ist nicht richtig mit dem Computer verbunden.

3. Der Netzteilstecker ist nicht richtig mit der Steckdose verbunden.

4. Die Stromzufuhr zu Ihrem Laufwerk ist gestört.

5. Die Diskettenlaufwerks-Schalter befinden sich nicht in der korrekten Position.

Wenn Sie alles überprüft und keine von diesen Fehlern gefunden haben, befolgen Sie die folgenden Schritte:

1. Benutzen Sie die DOS 2.5 Masterdiskette oder Systemdiskette (Arbeitskopie von DOS 2.5) um das System neu zu starten (Booten).

2. Entnehmen Sie die DOS - Diskette aus dem Laufwerk.

3. Legen Sie nun Ihre fehlerhafte Diskette in das Laufwerk ein, und sichern Sie die notwendigen Dateien auf einer anderen Diskette, indem Sie die Schritte für das Kopieren von Dateien ausführen. (siehe C.Kopieren von Dateien in Kapitel 3)

4. Mit der fehlerhaften Diskette in Laufwerk 1 benutzen Sie nun das Kommando "Delete File(s)" um alle Dateien auf der Diskette zu löschen.

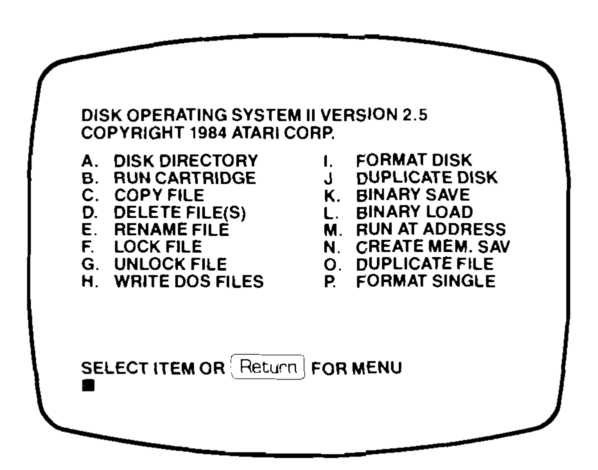
5. Versuchen Sie nun nochmals mit dieser Diskette zu arbeiten. Falls dies nicht klappt, sollten Sie diese Diskette nochmals formatieren (siehe I.,"Formatieren einer Diskette", in Kapitel 3).

Das DOS-Menü

Wenn Sie das DOS-Betriebssystem in Ihren Computer geladen haben, erscheint das DOS-Menue auf Ihrem Bildschirm. Dieses Menue ist eine Liste von DOS-Funktionen.

Die untere Zeile (Befehle) fordert Sie auf eine Auswahl zu treffen. Sie wählen die erforderliche Funktion, indem Sie den Buchstaben welcher dieser Funktion entspricht, eingeben und mit "Return" bestätigen.

DOS fragt dann die notwendigen Informationen um diese Funktion auszuführen ab (siehe "Befehle und Antworten" später in diesem Kapitel).



Es folgt eine Zusammenfassung des DOS 2.5 - Menues.

Die mit Sternchen (*) markierten Funktionen werden später in diesem Kapitel erklärt; alle werden im Kapitel 3 detailliert erklärt:

***A. Disk Directory**
Diese Funktion erlaubt Ihnen eine komplette oder einen Teil der Liste der auf Ihrer Diskette befindlichen Dateien auf dem Bildschirm auszugeben. Es werden Dateiname und Extender (wenn vorhanden), Anzahl der Sektoren für jede Datei und Anzahl der freien Sektoren auf der Diskette, angegeben.

***B. Run Cartridge**
(Diese Funktion kann nur mit eingebautem Basic benutzt werden oder wenn ein Steckmodul sich an dem dafür vorgesehenen Platz befindet.)
Diese Funktion erlaubt es Ihnen, die Kontrolle des Systems an das eingebaute Basic oder an das Steckmodul zurückzugeben (der linke Steckplatz im Atari 800 Computer, alle ATARI XL bzw. XE Computer verfügen über ein fest eingebautes BASIC).

***C. Copy File**

Benutzen Sie diese Funktion, wenn Sie zwei oder mehr Disketten-Laufwerke haben, und wenn Sie Dateien von einer Diskette auf eine andere Diskette kopieren wollen. Benutzen Sie diese Funktion auch wenn Sie eine Datei auf die gleiche Diskette kopieren wollen, indem Sie der Kopie einen anderen Namen zuweisen.

***D. Delete File(s)**

Mit dieser Funktion können Sie eine oder mehrere Dateien von Ihrer Diskette löschen. Dadurch bekommen Sie mehr Platz auf Ihrer Diskette.

E. Rename File

Benutzen Sie diese Funktion um einer vorhandenen Datei einen neuen Namen zu geben.

F. Lock File

Diese Funktion kann benutzt werden um zu verhindern, daß die Datei umbenannt wird, oder versehentlich gelöscht wird. Die Datei kann noch gelesen werden, eine Bearbeitung ist jedoch nicht mehr möglich. Wenn das Directory angezeigt wird, erscheint vor dem Dateinamen ein Stern. Dies bedeutet, daß diese Datei gesichert ist.

G. Unlock File

Diese Funktion entfernt den Stern vor dem Dateinamen. Somit können Sie den Namen der Datei wieder ändern, oder sie ganz löschen.

H. Write DOS-Files

Benutzen Sie diese Funktion um die DOS-Files (DOS.SYS und DUP.SYS) von Ihrer Systemdiskette auf eine andere Diskette zu kopieren.

***I. Format Disk**

Diese Funktion wird benutzt um eine leere Diskette zu formatieren. Dies ist unbedingt erforderlich, damit Sie Informationen auf dieser Diskette speichern können. Stellen Sie sicher das sich keine Dateien, welche noch gebraucht werden, auf dieser Diskette befinden. Diese Funktion formatiert eine Diskette mit großer Dichte, vorausgesetzt Sie benutzen ein XF551 oder 1050-Diskettenlaufwerk; ansonsten wird die Diskette in einfacher Dichte formatiert.

***J. Duplicate Disk**

Dies ist die Funktion, welche Sie brauchen um eine exakte Kopie Ihrer Diskette zu erstellen. Dabei wird die Zieldiskette automatisch formatiert.

K. Binary Save

Mit dieser Funktion können Sie den Inhalt spezifischer Speicherbereiche auf Diskette sichern. (Manipuliert Assembler-Programme.)

L. BINARY LOAD

Mit dieser Funktion können Sie Objekt-Dateien von Diskette wiederholen. (Manipuliert Assembler-Programme.)

M. RUN AT ADRESS

Mit dieser Option können Sie die hexadezimale Programm-Startadresse eingeben, nachdem Sie das Programm mit BINARY LOAD in die RAM geladen haben. (Führt Assembler-Programme aus.)

N. CREATE MEM.SAVE

Mit dieser Funktion können Sie auf einer Diskette Platz für ein Programm reservieren, welches sich gerade in der RAM befindet. Für einige Anwendungen, z.B. Programmieren, ist es empfehlenswert für jede neue Diskette, die als Systemdiskette verwendet werden soll, ein MEM.SAVE durchzuführen. Wenn Sie sich mit DOS eingearbeitet haben, werden Sie feststellen, daß bei einigen Anwendungen MEM.SAV-Dateien keine sinnvollen Funktionen erfüllen.

***O. DUPLICATE FILE**

Mit dieser Funktion können Sie Dateien von einer Diskette auf eine andere kopieren, selbst wenn Sie nur über ein Laufwerk verfügen.

***P. FORMAT SINGLE**

Benutzen Sie diese Funktion, um eine Diskette in einfacher Dichte mit einem XF551-Diskettenlaufwerk zu formatieren.

Prompts und Antworten

Die Fragen, welche DOS auf Ihrem Bildschirm anzeigt, werden Prompts genannt. DOS fragt Sie immer nach der Information, welche es benötigt um Ihren Wünschen gerecht zu werden.

Sie werden sich schnell in die meistbenutzten DOS-Prompts eingearbeitet haben. Da DOS für viele Funktionen die gleiche Information benötigt, wird es nicht lange dauern, bis Sie ganz automatisch antworten. Wenn Sie die DOS-Funktionen benutzen, listet das Programm nacheinander die Prompts und Antworten auf.

Nachdem Sie eine Antwort in den Computer eingegeben haben, müssen Sie die RETURN-Taste drücken, um Ihre Antwort zu bestätigen. (Wenn Sie nur die RETURN-Taste drücken wählt DOS einen DEFAULT aus - siehe DEFAULT, später in diesem Kapitel).

Viele Prompts verlangen eine einfache JA/NEIN - Antwort. Um JA zu antworten geben Sie einfach "Y" ein und bestätigen mit RETURN, um NEIN zu antworten geben Sie "N" ein und bestätigen wiederum mit RETURN.

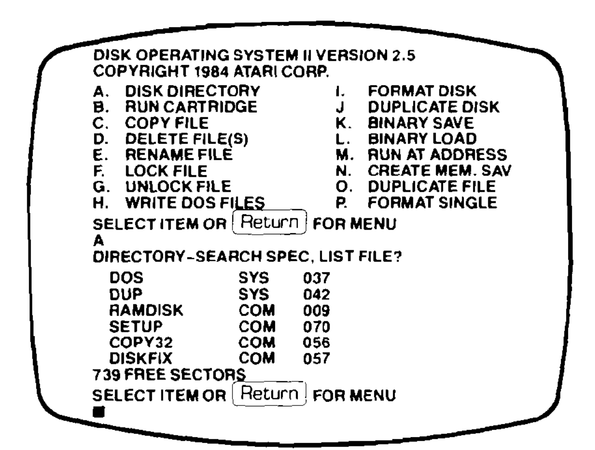
Wenn Sie einen Fehler während der Eingabe machen, drücken Sie DEL-Backspace um die Fehleingabe zu löschen. Dann geben Sie die korrekte Antwort ein. Um die ganze Eingabe zu löschen, bevor Sie mit RETURN bestätigt haben, geben Sie SHIFT/DEL-Backspace ein.

Das Disketten-Inhaltsverzeichnis

Jede Diskette, die Sie benutzen um Informationen zu speichern, hat ein Inhaltsverzeichnis, welches die abgespeicherten Dateien angibt, sowie deren Größe und wieviel Speicherplatz für weitere Informationen zur Verfügung steht.

Mit dem DISK DIRECTORY - Eintrag im DOS-Menue können Sie feststellen, welche Dateien sich auf Ihrer Diskette befinden.

Da auf der DOS-Masterdiskette sich Dateien befinden, können Sie diese Funktion dort ausprobieren. Mit dem DOS-Menue auf Ihrem Bildschirm brauchen Sie nur "A" einzugeben, und zweimal RETURN drücken.



Dies sind die Dateien des DOS 2.5 - Programmes. Die dreistelligen Nummern in der rechten Spalte zeigen an, wieviel Sektoren jede Datei auf der Master-Diskette belegt. (Siehe Formatieren einer Diskette für eine Erklärung der Sektoren). Die Zeile unter dem Index zeigt die noch freien Sektoren an, um weitere Informationen auf der Diskette zu speichern.

DOS.SYS und DUP.SYS sind Dateien, die Standard DOS-Funktionen ausführen. Für eine Erklärung von RAMDISK.COM siehe Anhang K. Für SETUP.COM, COPY32.COM und DISKFIX.COM siehe in Anhang L.

Kopieren von Disketten

Mit der DUPLICATE DISK - Funktion aus dem DOS 2.5 - Menue können Sie ein exaktes Duplikat einer Diskette herstellen. Diese Funktion kopiert alles von Ihrer Original(Source)-Diskette auf die Ziel(Destination)-Diskette. Dabei wird die Zieldiskette formatiert.

Warnung: Die DUPLICATE DISK - Funktion löscht alle Dateien, welche sich schon auf der Zieldiskette befinden. Benutzen Sie niemals eine Diskette als Zieldiskette, auf welcher sich wertvolle Dateien befinden.

Kopieren Ihrer DOS-Diskette

Um zu lernen wie die DUPLICATE-Funktion funktioniert, stellen Sie ein Duplikat Ihrer DOS 2.5 - Masterdiskette her. Dies ist eine wichtige Sicherungsmaßnahme. Sie sollten das Duplikat als Ihre Arbeitskopie benutzen, und die Masterdiskette als Sicherungskopie aufbewahren. Dann können Sie DOS benutzen, ohne sich Gedanken über versehentliches Löschen wichtiger Daten zu machen.

Als Zieldiskette benutzen Sie eine neue leere Diskette.

Wenn Sie nur ein Diskettenlaufwerk haben:

1. Mit dem DOS-Menue auf Ihrem Bildschirm geben Sie "J" ein, und bestätigen dies mit RETURN. Der Prompt "DUP DISK-SOURCE, DEST DRIVES ?" erscheint auf Ihrem Bildschirm.

2. Geben Sie 1,1 ein und bestätigen Sie mit RETURN. Der Prompt "INSERT SOURCE DISK, TYPE RETURN" erscheint.

3. Legen Sie die Diskette die Sie kopieren wollen in Ihr Laufwerk - in diesem Fall Ihre DOS 2.5 - Masterdiskette - und bestätigen Sie mit RETURN. Das Diskettenlaufwerk fängt nun an die Informationen auf Ihrer Diskette zu lesen. Dann fordert Sie DOS auf "INSERT DESTINATION DISK, TYPE RETURN", Ihre Zieldiskette einzulegen und mit RETURN zu bestätigen.

4. Entfernen Sie die Originaldiskette aus Ihrem Laufwerk und legen Sie die Zieldiskette ein (formatiert oder unformatiert), dann drücken Sie RETURN. DOS schreibt nun die Informationen, welche vorher von Ihrer Originaldiskette gelesen wurden, auf die Zieldiskette, wobei die Zieldiskette vorher formatiert wird.

Wieviel mal Sie DOS auffordert die Originaldiskette mit der Zieldiskette zu vertauschen, hängt davon ab wieviele Daten auf der Diskette gespeichert sind und wieviel RAM sich in Ihrem Computersystem befindet. Wenn der Prompt "SELECT ITEM OR RETURN FOR MENU" auf Ihrem Bildschirm erscheint, ist der Kopiervorgang abgeschlossen.

Beschriften Sie nun Ihre Arbeitskopie unter Verwendung der mitgelieferten Labels. Bringen Sie nun den Schreibschutz an Ihrer DOS-Arbeitskopie an (siehe: Der Schreibschutz). Bewahren Sie die DOS 2.5 - Masterdiskette an einem sicheren Ort auf.

Wenn Sie zwei Diskettenlaufwerke haben:

1. Mit dem DOS-Menue auf Ihrem Bildschirm geben Sie "J" ein und bestätigen dies mit RETURN. Der Prompt "DUP DISK - SOURCE, DEST DRIVES ?" erscheint auf Ihrem Bildschirm.

2. Geben Sie 1,2 ein und bestätigen Sie mit RETURN. Der Prompt "INSERT BOTH DISKS, TYPE RETURN" erscheint auf Ihrem Bildschirm.

3. Legen Sie Ihre Originaldiskette in Laufwerk 1 - in diesem Fall Ihre DOS 2.5 Masterdiskette - und eine leere Diskette (formatiert oder unformatiert) in Laufwerk 2 und bestätigen mit RETURN.

4. DOS kopiert nun alle Informationen von der Originaldiskette auf die Zieldiskette, wobei die Zieldiskette automatisch formatiert wird.

Wenn der Prompt "SELECT ITEM OR RETURN FOR MENU" erscheint, ist der Kopiervorgang abgeschlossen.

Beschriften Sie nun Ihre Arbeitskopie unter Verwendung der mitgelieferten Labels. Bringen Sie nun den Schreibschutz an Ihrer DOS-Arbeitskopie an (siehe: Der Schreibschutz). Bewahren Sie die DOS 2.5 - Masterdiskette an einem sicheren Ort auf.

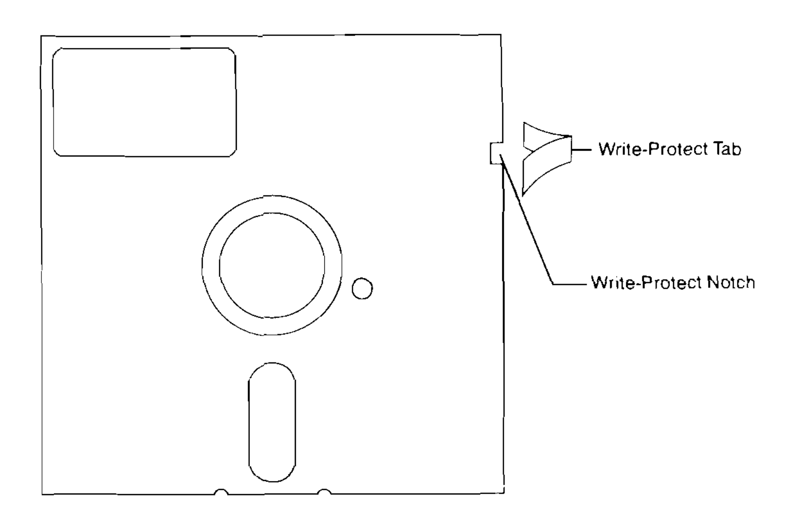
(Eine andere Möglichkeit die DOS Systemdiskette zu kopieren ist die Option "H.WRITE DOS FILES", um DOS.SYS und DOS.DUP auf eine formatierte Diskette zu schreiben. Bei einigen Anwendungen, z.B. AtariWriter ist es angebracht wenn Sie DOS-Dateien auf jede Diskette schreiben, die Sie benutzen um Dateien abzuspeichern. Auf diesem Weg können Sie DOS direkt von Ihrer Datendiskette laden, während Sie ein Applikationsprogramm laden (siehe auch Kap. 3).

Die DUPLICATE DISK Funktion wird manchmal mit der COPY FILES Funktion des DOS verwechselt. Die COPY FILES Funktion kopiert nur die angegebenen Dateien von Ihrer Originaldiskette. Die DUPLICATE DISK Funktion arbeitet effizienter wenn Sie eine komplette Sicherungskopie von einer Datendiskette mit mehreren Dateien erstellen wollen.

Der Schreibschutz

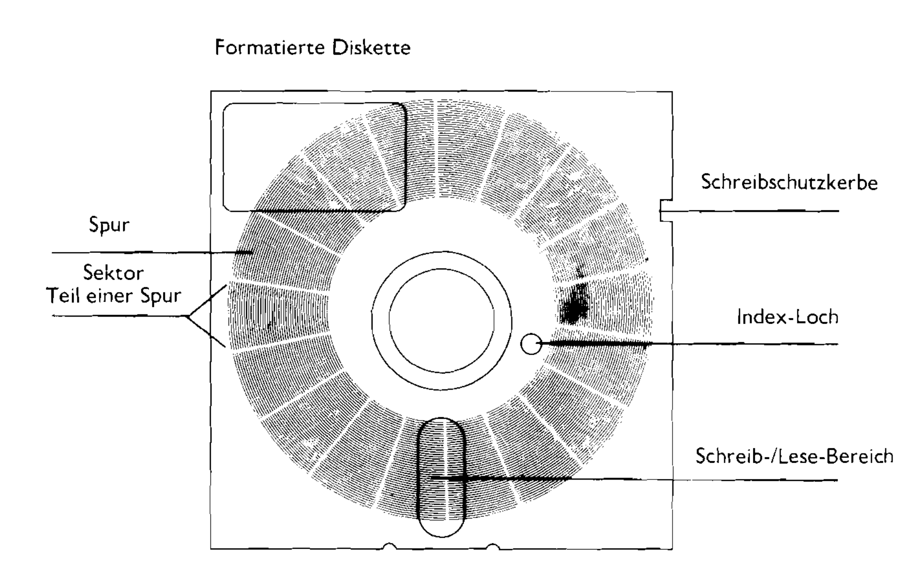
Bevor Sie eine Diskette kopieren ist es empfehlenswert den Schreibschutz an der Originaldiskette anbringen. (In jeder Diskettenpackung die Sie kaufen, werden selbstklebende Streifen mitgeliefert.) Wenn Sie die Einkerbung an der Kante der Diskette mit diesem Klebestreifen zukleben, kann auf diese Diskette keine Information mehr geschrieben werden.

Insbesondere wenn Sie nur ein Diskettenlaufwerk benutzen um Disketten zu kopieren, kann es passieren das Sie aus Versehen die Originaldiskette in Ihr Laufwerk einlegen, wenn das System gerade Daten auf Ihre Zieldiskette schreiben will. Der Schreibschutz verhindert in diesem Fall das Zerstören wichtiger Daten auf Ihrer Originaldiskette.



Formatieren einer Diskette

Ihre gekauften Disketten müssen zunächst einmal darauf vorbereitet werden, Daten aufzunehmen. Dieser Prozess wird Formatieren oder Initialisieren einer Diskette genannt. Formatieren organisiert die Oberfäche einer Diskette in Spuren und Sektoren, sodaß der Computer Informationen abspeichern und Lesen kann (siehe Illustration).



Warnung: Formatieren einer Diskette löscht alle Daten, welche sich bereits auf dieser Diskette befinden. Formatieren Sie niemals Ihre Masterdiskette oder eine Datendiskette mit wertvollen Dateien.

Das DOS 2.5 Menu stellt Ihnen zwei Optionen zum Formatieren von Disketten zur Verfügung. Wenn Sie ein XF551 Laufwerk benutzen, formatiert die Option "I.FORMAT DISK" Ihre Diskette in großer Dichte. Wenn Sie ein 810 Laufwerk benutzen, formatiert das DOS Ihre Diskette in einfacher Dichte. Die Option "P.FORMAT SINGLE" formatiert Ihre Diskette immer in einfacher Dichte. Sie sollten diese Funktion benutzen, wenn Sie mit einem XF551 Laufwerk eine Diskette einseitig formatieren wollen.

Wenn Ihr System beide Laufwerke, ein Atari XF551 und 810 Laufwerk beinhaltet, oder wenn Sie Dateien unter dem früheren Atari DOS 2.0 erstellt haben, siehe Anhang H. Wenn Sie Dateien mit DOS 3 erstellt haben, siehe Anhang L.

Wie man eine Diskette formatiert

Um sich mit dem Vorgang des Formatierens vertraut zu machen, formatieren Sie zunächst zwei leere Disketten, die Sie für die folgende Übung in diesem Kapitel benützen. Mit dem DOS Menu auf Ihrem Bildschirm:

1. Um eine Diskette in großer Dichte auf einem XF551 Laufwerk zu formatieren geben Sie "I" ein und bestätigen dies mit RETURN; oder um in einfacher Dichte zu formatieren geben Sie "P" ein und bestätigen mit RETURN. Der Prompt "WHICH DRIVE TO FORMAT ?" erscheint.

2. Geben Sie die Nummer des Laufwerks mit dem Sie die Diskette formatieren wollen ein. Der Prompt "TYPE "Y" TO FORMAT DISK n" (wobei n die Nummer des von Ihnen ausgewählten Laufwerks ist) erscheint. Das gibt Ihnen die Möglichkeit sicherzustellen, daß das von Ihnen ausgewählte Laufwerk die zu formatierende Diskette enthält - Denken Sie daran daß das Formatieren alle Dateien auf der Diskette löscht.

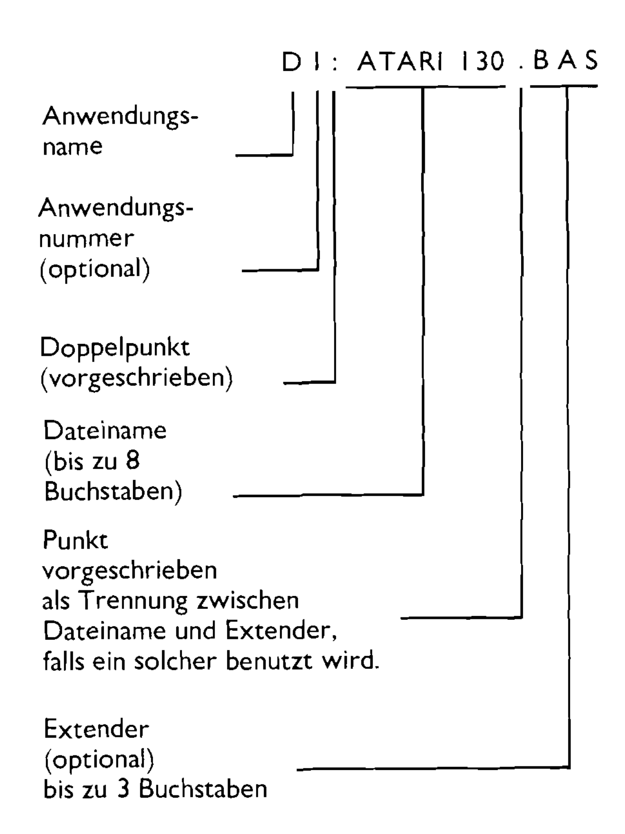
3. Legen Sie die Diskette in das Laufwerk, welches Sie angegeben haben, ein. Geben Sie "Y" ein und bestätigen mit RETURN. Die Kontrolleuchte an Ihrem Laufwerk leuchtet auf, sobald der Formatierungsprozess beginnt.

Wenn der Prompt "SELECT ITEM OR RETURN FOR MENU" erscheint, ist der Formatierungsprozess abgeschlossen. Sie können nun diese formatierte Diskette benutzen um Dateien oder DOS-Dateien abzuspeichern.

Wiederholen Sie die oben beschriebenen Schritte um Ihre zweite Diskette zu formatieren.

Benennen und Arbeiten mit Dateien

Um mit Dateien unter DOS zu arbeiten, müssen Sie jeder Datei einen Dateinamen geben. Weiterhin müssen Sie einen Anwendungscode benutzen um Ihrem Computer mitzuteilen welcher Teil - z.B. Laufwerk 1 - angesprochen werden soll, um Ihre Datei zu handhaben. Zusammengenommen, der Anwendungscode und der Dateiname den Sie angegeben haben, ergibt ein filespec (Dateispezifikation). So sieht eine typischer filespec aus:



Anwendungs-Codes

Das D1: in der filespec - Illustration ist der Anwendungs-Code; dort repräsentiert es den Teil des Systems wo DOS den Befehl ausführen soll. Das "D" steht für Laufwerk und die "1" gibt die Nummer des Laufwerks in Ihrem Computer an. Der Doppelpunkt muß immer mit einem Anwendungscode benutzt werden.

Es gibt also noch andere Codes für Geräte die DOS benutzen oder aktivieren kann. Das Default Anzeige-Gerät zum Beispiel (siehe Defaults) ist "E:". Es steht für TV-Bildschirm oder Monitor. Sie können auch "P:" benutzen, welches für Drucker steht. Sehen Sie z.B. unter DISK DIRECTORY in Kapitel 4 für Beispiele um verschiedene Anzeigegeräte zu benutzen. "C:" steht für Casettenrecorder. Dies ist ein anderer Anwendungscode, den Sie vielleicht angeben wollen, wenn Sie DOS benützen.

Defaults

Der Einfachheit halber sind in DOS 2.5 Default-Antworten vorgegeben.

Da die meisten DOS-Benutzer nur ein Laufwerk besitzen ist eine der bequemeren Defaults in DOS "D1:" - Laufwerk 1 in Ihrem Computersystems. Sie haben schon gesehen wie DOS diesen Default bei den Prompts "DIRECTORY SEARCH SPEC, LIST FILE" benutzt, wenn Sie RETURN drücken. DOS benutzt auch den Default nach dem "D1:" als Quelle und Zielgerät - vorausgesetzt das Sie RETURN als Antwort zu einem geeigetem Prompt eingegeben haben.

Wenn Sie nur ein Laufwerk besitzen, brauchen Sie die Laufwerksnummer innerhalb des Anwendungscodes nicht anzugeben. DOS versteht "D:" als Laufwerk 1.

Dateinamen

Jeder Datei, die Sie auf einer Diskette abspeichern wollen, muß ein unverwechselbarer Dateiname zugewiesen werden. Ansonsten würde Ihr Computer nicht wissen, mit welcher Datei Sie arbeiten wollen.

Jeder Dateiname darf eine Länge von bis zu acht Buchstaben haben, gefolgt, wenn Sie wollen, von einem Punkt und einem Extender, welcher bis zu drei Buchstaben umfassen darf. Außer dem Punkt, der den Extender vom Dateinamen trennt, dürfen alle Buchstaben und Nummern verwendet werden, jedoch keine Sonderzeichen oder andere Symbole, z.B.:

Richtig:	**Falsch:**
PROGRAM.6J	PROG.6J.BAS
ACCT4321	ACCOUNT4321
LETTER1	

Wenn Sie versuchen einen falschen Dateinamen einzugeben, gibt das DOS "ERROR 165" auf Ihrem Bildschirm aus, und akzeptiert diesen Befehl nicht.

Extender, manchmal Dateitypen genannt, können bei der Namensgebung sehr nützlich sein, wenn Sie miteinander verwandte Dateien in Gruppen zusammenfassen wollen (siehe Wildcards). So können Sie z.B. den Extender ".BAS" benutzen, um alle Programme die Sie in AtariBasic geschrieben haben zu benennen - PROGRAM1.BAS, PROGRAM2.BAS etc.

Wildcards

Wildcards (Joker) werden vom DOS 2.5 genau wie Joker in einem Pokerspiel benutzt. Sie stehen für jede Karte. DOS berücksichtigt diese Wildcard-Symbole für jeden Buchstaben oder Kombination von Buchstaben in einem Dateinamen. Der größte Vorteil bei Benutz-ung von Wildcards liegt darin, daß Sie eine Gruppe von Dateien handhaben können.

Die zwei Wildcards, welche DOS erkennt, sind das Fragezeichen (?), welches für einzelne Buchstaben steht und der Stern (*), welcher für eine Kombination von Buchstaben in einem Dateinamen oder Extender steht. Zum Beispiel beim Arbeiten mit den folgenden Dateien könnten Sie also "PROGRAM?.*" angeben, um alle Programme zu spezifizieren.

PROGRAM1.BAS
PROGRAM2.BAS
PROGRAM3.BAS

Jetzt verstehen Sie die ganzen Default filespecs, welche von DOS bei einer DISK DIRECTORY-Prozedur benutzt werden. Wenn Sie beim Prompt "DIRECTORY-SEARCH SPEC,LIST FILE" mit Return bestätigen, versteht es DOS als "D1:*.*". Das "D1:" ist der Defaultcode für Laufwerk 1. Das "*.*" steht für alle Dateien auf Ihrer Diskette.

Laden eines Steckmoduls unter DOS

Mit AtariBasic (ob in Steckmodul-Form oder im Computer eingebaut) oder jeder anderen Steckmodul-Programmiersprache können Sie Ihre eigenen Programme schreiben und auf Ihrem Atari-Computer laufen lassen. Weiterhin können Sie Programmiersprachen-Kommandos benutzen um Programme auf Diskette abzuspeichern oder von Diskette zu laden. Sie brauchen aber DOS für alle anderen Anwendungen auf Ihre Programmdateien - z.B. Kopieren oder Löschen einer Datei. Die Funktion RUN CARTRIDGE aus Ihrem DOS-Menu erlaubt Ihnen die Kontrolle von DOS an das Steckmodul zu übergeben, um somit z.B. das AtariBasic aufzurufen.

Von BASIC nach DOS und zurück

Wenn Sie BASIC und DOS zusammen laden, wie schon in einem früheren Kapitel beschrieben, erscheint der Prompt "READY" auf Ihrem Bildschirm. Geben Sie einfach den Befehl "DOS" ein um von Basic zu DOS zu gelangen. Das DOS-Menu erscheint auf Ihrem Bildschirm.

Um von DOS nach Basic zu gelangen (oder jeder anderen Steckmodul-Programmiersprache) geben Sie "B" (RUN CARTRIDGE) ein und bestätigen Sie mit RETURN.

Sichern und Laden eines BASIC-Programmes

Die Basic-Computersprache beinhaltet Ihre eigenen Sichern- und Ladeninstruktionen, die Sie benutzen können um ein Programm von einer Diskette zu laden oder zu sichern. Um ein Basic-Programm zu sichern oder zu laden müssen Sie zuerst RUN CARTRIDGE vom DOS-Menu selektieren. Wenn der Prompt READY auf Ihrem Bildschirm erscheint, geben Sie das folgende Programm genauso wie es hier erscheint (selbst Leerzeichen und Satzzeichen sind von entscheidender Bedeutung beim Programmieren) ein. Bestätigen Sie mit RETURN am Ende einer jeden Zeile. Wenn Sie einen Fehler gemacht haben, können Sie diesen mit DEL/Backspace korrigieren.

```
10 PRINT "DIESE ZEILE WIEDERHOLT SICH"
20 GOTO 10
```

Damit haben Sie ein zweizeiliges Basic-Programm geschrieben, welches dem Computer mitteilt, "DIESE ZEILE WIEDERHOLT SICH" auf dem Bildschirm auszugeben und danach zu der ersten Instruktion zurückzuspringen. Sie können sich vorstellen was geschieht wenn dieses Programm läuft.

Versuchen Sie es!

Geben Sie "RUN" ein, und bestätigen Sie mit RETURN. Wenn Sie genug haben, drücken Sie die "BREAK-Taste" um das Programm anzuhalten.

Um dieses oder andere Programme auf Diskette abzuspeichern müssen Sie den entsprechenden BASIC-Befehl - SAVE - gefolgt von einer filespec die DOS versteht eingeben. Tauschen Sie Ihre DOS-Diskette in Laufwerk 1 gegen die von Ihnen formatierte Leerdiskette aus. Nun geben Sie " SAVE "D:PROGRAM-1.BAS" " ein und bestätigen mit RETURN. Sowie Ihr Diskettenlaufwerk zu arbeiten beginnt, wird Ihr Programm auf Diskette abgespeichert.

Obwohl Ihr Programm auf Diskette gespeichert ist, bleibt es trotzdem im rechner-internen Speicher erhalten. Um zu sehen wie das Programm wieder in den Computer geladen wird, müssen Sie den Befehl "NEW" eingeben und mit RETURN bestätigen (der rechner-interne Speicher wird gelöscht). Jetzt drücken Sie Shift/CLEAR um den Text von Ihrem Bildschirm zu löschen. Danach geben Sie " LOAD "D:PROGRAM1.BAS" " ein und bestätigen mit RETURN. Jetzt wird das Programm von Ihrem Laufwerk zurück in den Computer geladen. Zuletzt geben Sie "LIST" ein und bestätigen mit RETURN um das Programm auf dem Bildschirm wieder sichtbar zu machen.

Bemerkung: Die Basic LOAD und SAVE-Befehle sind nicht die gleichen wie DOS LOAD und SAVE.

Obwohl Sie jetzt das Programm im rechner-internen Speicher haben, bleibt es trotzdem auf Ihrer Diskette. Eine einmal gesicherte Datei bleibt solange auf Diskette erhalten, bis Sie den Befehl "DELETE FILE (S)" aus Ihrem DOS-Menu aufrufen (siehe Löschen von Dateien später in diesem Kapitel).

Kopieren von Dateien

Mit den Befehlen "COPY FILE" und "DUPLICATE FILE" aus Ihrem DOS-Menu können Sie Dateien von einer Diskette zu einer anderen Diskette kopieren. Sie können auch eine Backup-Kopie auf der gleichen Diskette erzeugen, indem Sie der Kopie Ihrer Datei einen anderen Dateinamen geben.

Bemerkung: Sie können die DOS-Files DOS.SYS und DUP.SYS nicht mit den Befehlen COPY FILE oder DUPLICATE FILE kopieren. Sie müssen stattdessen den Befehl "H WRITE DOS FILE" benutzen um diese Dateien zu kopieren (siehe Kapitel 3).

Erstellen einiger Übungs-Dateien

Während Sie die vorhergehenden Kapitel in diesem Handbuch durchgearbeitet haben, haben Sie ein kleines Basic-Programm geschrieben und es auf Ihrer Diskette unter dem Namen PROGRAM1.BAS abgespeichert. Um zu lernen wie die Kopierfunktionen benutzt werden, werden Sie jetzt noch einige Übungsdatei-en erstellen.

Laden Sie DOS und BASIC, wenn erforderlich. Wenn Sie dies schon gemacht haben, erscheint das DOS-Menu auf Ihrem Bildschirm. Wählen Sie RUN CARTRIDGE. Legen Sie Ihre Datendiskette in das Laufwerk I (die Diskette mit der Datei PROGRAM1.BAS). Wenn der READY-Prompt auf Ihrem Bildschirm erscheint, geben Sie die unten aufgeführten drei SAVE-Kommandos ein. Drücken Sie RETURN nach jeder Zeile und warten Sie bis Ihr Diskettenlaufwerk die Datei gesichert hat, bevor Sie weiterarbeiten. Im Grunde genommen bestehen diese Dateien nur aus Dateinamen, aber Sie reichen aus um mit ihnen zu lernen wie man Dateien kopiert.

```
SAVE "D:PROGRAM2.BAS"
SAVE "D:PROGRAM3.BAS"
SAVE "D:PROGRAM1.PIL"
```

Dateien kopieren

Welchen der beiden Kopierbefehle Sie benutzen, hängt davon ab wieviele Laufwerke Sie besitzen und ob Sie eine Datei von einer Diskette auf eine andere, oder auf die gleiche Diskette kopieren wollen. Wenn Ihr System ein Atari XF551-Laufwerk und ein 810 Laufwerk beinhaltet, oder wenn Sie Dateien mit dem früheren Atari DOS 2.0S erstellt und abgespeichert haben, sehen Sie unter Anhang H. Wenn Sie DOS 3 - Dateien haben, sehen Sie unter Anhang L nach.

Von einer Diskette auf eine andere mit einem Laufwerk

Befolgen Sie folgende Schritte um eine Datei von einer Diskette auf eine andere mit einem Laufwerk zu kopieren:

1. Mit dem DOS-Menu auf dem Bildschirm geben Sie "O" für DUPLICATE FILE ein, und bestätigen Sie mit RETURN. Der Prompt "NAME OF FILE TO MOVE?" erscheint auf Ihrem Bildschirm.

2. Geben Sie den Namen einer Ihrer Übungsdateien - zB. PROGRAM1.PIL - ein, und bestätigen Sie mit RETURN. Der Prompt "INSERT SOURCE DISK, TYPE RETURN" erscheint.

3. Legen Sie die Diskette mit der angegebenen Datei, welche Sie kopieren wollen, in Ihr Laufwerk und drücken Sie RETURN. Das Laufwerk fängt nun an die angegebene Datei von Ihrem Laufwerk zu lesen. Das DOS fordert Sie nun mit dem Prompt "INSERT DESTINATION DISK, TYPE RETURN" auf, die Zieldiskette einzulegen.

4. Entfernen Sie nun die Originaldiskette aus Ihrem Laufwerk und legen Sie eine formatierte Diskette ein, dann drücken Sie RETURN.

Wieviel mal Sie das DOS auffordert die Originaldiskette mit der Zieldiskette zu vertauschen, hängt von der Größe der Original-Datei ab. Wenn der Prompt "SELECT ITEM OR RETURN FOR MENU" erscheint, ist der Kopiervorgang abgeschlossen.

Von einer Diskette zu einer anderen mit zwei Laufwerken

1. Mit dem DOS-Menu auf Ihrem Bildschirm geben Sie "C" für COPY FILE ein und drücken RETURN. Der Prompt "COPY-FROM, TO?" erscheint.

2. Geben Sie den kompletten Pfadnamen der Datei, welche Sie kopieren wollen ein, sowie den Pfadnamen der Kopie. Für die Übungsdatei PROGRAM1.PIL geben Sie folgendes ein:
   D1:PROGRAM1.PIL,D2:PROGRAM.PIL

3. Vergewissern Sie sich das die Diskette mit der Originaldatei sich in Laufwerk 1 befindet, und die Zieldiskette in Laufwerk 2. Dann drücken Sie RETURN.

Nun kopiert DOS die angegebene Datei in Laufwerk 1 nach Laufwerk 2. Wenn der Prompt "SELECT ITEM OR RETURN FOR MENU" erscheint, ist der Kopiervorgang abgeschlossen.

Sicherungskopie einer Datei auf der gleichen Diskette

Egal ob Sie ein oder zwei Laufwerke besitzen, benutzen Sie COPY FILE aus dem DOS-Menu um eine Sicherungskopie einer Datei auf derselben Diskette anzufertigen. Befolgen Sie die Schritte unter "Von einer Diskette auf die andere mit zwei Laufwerken", aber geben Sie das gleiche Laufwerk für beide Pfadnamen an und vergessen Sie nicht der Zieldatei einen anderen Namen als der Originaldatei zu geben. Zum Beispiel, wenn Sie Ihre Übungsdatei PROGRAM1.PIL kopieren:

D1:PROGRAM1.PIL,D1:PROGRAM1.BAK

Benutzen von Wildcards um eine Gruppe von Dateien zu kopieren

Nehmen wir einmal an, Sie wollen Sicherungskopien von Ihren vier Übungsdateien anfertigen - PROGRAM1.BAS, PROGRAM2.BAS, PROGRAM3.BAS und PROGRAM1.PIL . Sie können Wildcards benutzen, um alle vier Dateien auf einmal zu kopieren - eine zeitsparende Alternative.

Um Wildcards für den Kopiervorgang zu benutzen, befolgen Sie die gleiche Prozedur wie beim Kopieren einer Datei, aber benutzen Sie Wildcards für den Pfadnamen. Um alle vier Ihrer Übungsdateien zu kopieren, würden Sie statt dem Dateinamen, PROGRAM?.* benutzen - das Fragezeichen steht für die Nummern in allen vier Dateinamen und der Stern steht für die Extender in allen vier Dateinamen. Wenn Sie den DUPLICATE FILE - Befehl benutzen, teilt Ihnen DOS mit wenn eine Datei kopiert wurde.

Sehen Sie unter Kapitel 3 für detaillierte Beispiele für die Benutzung von Wildcards, wenn Sie die Kopierbefehle COPY FILE und DUPLICATE FILE um Dateien zu kopieren benutzen.

Löschen von Dateien

Sie können eine Datei mit dem Befehl DELETE FILE(S) aus dem DOS-Menu von Ihrer Diskette löschen. Das Löschen von veralteten Dateien ermöglicht es Ihnen mehr Platz für neue Dateien zu schaffen. Nachdem eine Datei von der Diskette gelöscht wurde, verschwindet der Dateiname aus dem Disketten-Inhaltsverzeichnis (Directory).

Warnung: Benutzen Sie DELETE FILE(S) mit Vorsicht - dieser Befehl ist nicht mehr rückgängig zu machen. Nachdem eine Datei gelöscht wurde, kann sie nur unter bestimmten Voraussetzungen mit DISKFIX.COM wiedergeholt werden-.(siehe unter dem Abschnitt DISKFIX.COM im Anhang L nach)

Aus Übungszwecken versuchen Sie nun die Kopie von der Datei PROGRAM1.BAS zu löschen. Legen Sie Ihre Datendiskette mit dieser Datei in das Laufwerk. Dann befolgen Sie die weiteren Schritte:

1. Mit dem DOS-Menu auf Ihrem Bildschirm geben Sie "D" ein und bestätigen mit RETURN. Der Prompt "DELETE FILESPEC" erscheint.

2. Geben Sie "D1:PROGRAM1.BAS" ein, wenn Ihre Datendiskette sich im Laufwerk 1 befindet, oder "D2:PROGRAM1.BAS" wenn sie sich in Laufwerk 2 befindet. Danach drücken Sie RETURN. Der Prompt "TYPE "Y" TO DELETE..." erscheint - diese nochmalige Abfrage erlaubt es Ihnen nochmals zu überdenken ob Sie diese Datei wirklich löschen wollen.

3. Geben Sie "Y" ein und RETURN um diese Datei zu löschen.

Mit Wildcards in Ihrem Dateinamen haben Sie die Möglichkeit mehrere Dateien mit einer Operation zu löschen. Um dies auszuprobieren, löschen Sie die zwei verbliebenen Kopien der Dateien mit dem BAS - Extender auf Ihrer Diskette. Befolgen Sie die weiteren Schritte, welche Sie benutzt haben um eine Datei zu löschen, aber statt des Dateinamens benutzen Sie "*.BAS". Dann sieht der Befehl wenn der Prompt "DELETE FILESPEC" erscheint wie folgt aus:

D1:*.BAS (oder D2:*.BAS wenn Sie Laufwerk 2 benutzen)

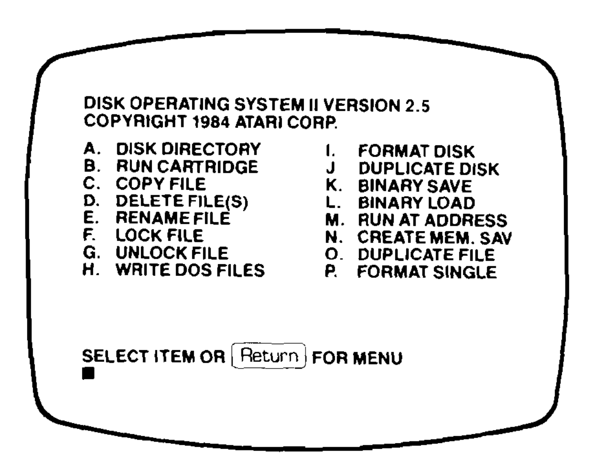
Das DOS gibt nun jeden Dateinamen der gelöscht werden soll aus. Geben Sie "Y" ein und bestätigen Sie mit RETURN um die jeweilige Datei zu löschen. Wenn Sie mehrere, jedoch nicht alle Dateien die Sie mit Wildcards angegeben haben, löschen wollen, geben Sie einfach "N" ein und drücken RETURN für den Namen der Datei die Sie nicht löschen wollen. Diese Operation ist schneller als wenn Sie für jede Datei die "DELETE FILE(S)" Prozedur einzeln aufrufen.

Kapitel 3: Auswählen eines DOS-Menue-Befehls

Wie Sie einen DOS-Menue-Befehl auswählen:

1. Laden Sie DOS in Ihren Computer. (Vom BASIC aus gelangen Sie nach DOS, indem Sie den Befehl "DOS" eingeben und die Return-Taste drücken.

2. Das DOS-Menue mit seinen 16 Einträgen erscheint auf Ihrem Bildschirm.



3. Geben Sie den Buchstaben ein, der Ihrer Wahl entspricht und bestätigen Sie mit RETURN.

4. Ein Prompt, der für das DOS notwendige Information abfragt, erscheint auf Ihrem Bildschirm. Diese Informationen (Parameter) geben an, wie der Befehl ausgeführt wird (manchmal optional).

5. Der Prompt "SELECT ITEM OR RETURN FOR MENU" erscheint immer dann auf Ihrem Bildschirm, wenn das Computer-System einen Befehl ausgeführt hat. Wenn Sie weitere Befehle ausführen wollen, geben Sie den entsprechenden Buchstaben ein und bestätigen Sie mit RETURN. Die untere Hälfte des Bildschirms wird nach oben gerückt, um für die Prompts des nächsten Befehls Platz zu machen. Wenn Sie nur die Return-Taste drücken, wird der Bildschirm gelöscht und dann das DOS-Menue angezeigt.

Wenn Sie in Ihrem System gleichzeitig ein ATARI XF551 und ein 810 Laufwerk haben, oder wenn Sie Dateien, die mit ATARI DOS 2.0S erstellt wurden haben, dann sehen Sie im Anhang H nach für detaillierte Informationen um mit Dateien und Disketten unter DOS 2.5 umzugehen. Wenn Sie Dateien, die mit DOS 3 formatiert wurden besitzen, sehen Sie im Anhang L.

A. Disketten-Inhaltsverzeichnis (Disk Directory)

Ein Disketten-Inhaltsverzeichnis (Disk Directory) ist eine Liste der Dateien, die sich auf der Diskette befinden. Es werden weiterhin der Dateiname, der Extender (wenn vorhanden) und die Anzahl der Sektoren, die die Datei belegt, angezeigt. Dabei wird entweder ein Teil oder die ganze Liste der Dateien angezeigt, je nachdem welche Parameter eingegeben wurden. Wildcards können als Parameter benutzt werden.

Wenn der Prompt "SELECT ITEM OR RETURN FOR MENU" auf Ihrem Bildschirm ist, geben Sie "A" ein und drücken Sie die Return-Taste. Auf Ihrem Bildschirm erscheint folgende Mitteilung:

```
DIRECTORY--SEARCH SPEC,LIST FILE?
```

Wenn Sie jetzt "RETURN" drücken, dann erscheint das komplette Inhaltsverzeichnis Ihrer Diskette, mit allen Dateinamen, mit der Größe der Datei (in Sektoren) und die Anzahl der noch freien Sektoren auf der Diskette. Das folgende Beispiel zeigt Ihnen die Dateien des Inhaltsverzeichnis Ihrer DOS 2.5 System Diskette.

```
SELECT ITEM OR RETURN FOR MENU
A [RETURN]
DIRECTORY--SEARCH,LIST FILE? [RETURN]
```

```
   DOS SYS 037
   DUP SYS 042
   RAMDISK COM 009
   SETUP COM 070
   COPY32 COM 056
   DISKFIX COM 057
739 FREE SECTORS

SELECT ITEM OR RETURN FOR MENU
```

In dem Fall, daß Sie einmal eine Datei, die 1000 oder mehr Sektoren groß ist, mit DOS 2.5 abspeichern wollen, zeigt Ihnen das Disketten-Inhaltsverzeichnis für diese Datei nur eine Größe von 999 Sektoren an. Ähnlich, wenn auf der Diskette mehr als 999 Sektoren unbenutzt sind, zeigt das Inhaltsverzeichnis "999 + FREE SECTORS" an. Dies ist kein Fehler. Um eine Kompatibilität zu anderen DOS-Versionen zu erhalten (DOS 2.0S), benutzt DOS 2.5 nur dreistellige Zahlen um Dateigrößen anzuzeigen.

Wenn Sie Dateien abspeichern, benutzt DOS 2.5 die untersten Sektoren einer Diskette zuerst. Da das alte DOS 2.0S nur ein Maximum von 719 Sektoren anspricht, kann dieses auch keine Dateien benutzen, die mit DOS 2.5 erstellt wurden und Sektoren 720 und größer belegen. Wenn Sie die Option "A" benutzen, gibt DOS 2.5 diese Dateien in Klammern aus. Zum Beispiel:

```
  DOS SYS 037
  DUP SYS 042
  FILE1 DAT 204
  FILE2 DAT 119
 <FILE3 DAT> 350
 <FILE4 DAT> 022
```

In diesem Inhaltsverzeichnis werden die beiden Dateien FILE3.DAT und FILE-4.DAT mit Klammern dargestellt, da sie Sektoren 720 und größer auf der Diskette belegen. Beachten Sie, daß das nicht von der größe der Datei abhängt. Weiterhin werden Dateien, die diese Sektoren belegen, normalerweise, aber nicht immer, als letzte Dateien im Inhaltsverzeichnis angezeigt.

Parameter für den Disk-Directory-Befehl

Wie Sie schon vom Eingabe-Prompt des Inhaltsverzeichnis erkennen, hat dieser Befehl zwei Eingabeparameter: SEARCH SPEC und LIST FILE.

Bei diesem Prompt können Sie wählen zwischen Suchen einer einzelnen Datei, mehreren Dateien oder allen Dateien, die Sie angeben. Wenn Sie kein spezifisches Laufwerk angeben, nimmt das DOS an, daß Sie die Dateien von Laufwerk 1 sehen wollen Default-Laufwerk).

Wenn Sie keinen speziellen Dateinamen angeben, benutzt DOS den Default-Wert "D1:*.*,E:" für beide Parameter. Der erste Default-Parameter, D1:*.*, teilt DOS mit, daß Sie eine Liste aller Dateien, die sich auf der Diskette in Laufwerk 1 befinden, sehen wollen.

Der zweite Default-Parameter, E:, teilt DOS mit, daß Sie diese Information auf dem Bildschirm ausgegeben haben wollen. Aus diesem Grund, wenn Sie keinen der beiden Parameter angeben, zeigt DOS alle Dateinamen mit Dateigröße an; für die Diskette, die sich in Laufwerk 1 befindet.

Wenn Sie einen Drucker besitzen, können Sie sich das Inhaltsverzeichnis auch ausdrucken lassen, indem Sie ein Komma (,) nach dem ersten Parameter benutzen und "P:" für den zweiten. In dem folgenden Beispiel wird nur die Information für eine Datei auf dem Drucker ausgegeben:

1. Geben Sie "A" ein und drücken Sie [RETURN].

2. Nachdem der Eingabe-Prompt erscheint, geben Sie "DOS.SYS,P:" ein und drücken [RETURN].

3. Jetzt wird ein Teil des Inhaltsverzeichnis auf Ihrem Drucker ausgegeben, anstatt auf Ihrem Bildschirm angezeigt. Wenn Sie keinen Drucker angeschlossen haben oder wenn er nicht eingeschaltet ist, dann wird "ERROR 138" auf Ihrem Bildschirm angezeigt.

Auf der Kopie, die der Drucker angefertigt hat werden Sie folgendes sehen:

```
  DOS SYS 037
739 FREE SECTORS
```

Jedesmal wenn der DISK DIRECTORY-Befehl seine Arbeit verrichtet hat, erscheint der Prompt "SELECT ITEM OR [RETURN] FOR MENU" auf Ihrem Bildschirm.

In den folgenden Beispielen werden einige unterschiedliche Möglichkeiten für diesen Befehl dargestellt.

Bemerkung: Wenn Dateinamen angezeigt werden, dann wird der Dateiname und dessen Extender nicht durch einen Punkt getrennt. Wenn Sie jedoch auf eine Datei Bezug nehmen, dann müssen immer den Dateinamen vom Extender mit einem Punkt trennen.

Beispiel 1: Gibt eine Liste aller Dateien von der Diskette in Laufwerk 1 mit dem Extender ".SYS" aus.

```
SELECT ITEM OR [RETURN] FOR MENU
A [RETURN]
DIRECTORY--SEARCH SPEC,LIST FILE?
*.SYS [RETURN]
```

Beispiel 2: Gibt alle Dateien der Diskette in Laufwerk 2 auf dem Drucker aus.

```
SELECT ITEM OR [RETURN] FOR MENU
A [RETURN]
DIRECTORY--SEARCH SPEC,LIST FILE?
D2:,P: [RETURN]
```

Beispiel 3: Gibt alle Dateinamen der Diskette in Laufwerk 1 mit drei Buchstaben, die mit EO anfangen, auf Ihrem Bildschirm aus.

```
SELCET ITEM OR [RETURN] FOR MENU
A [RETURN]
DIRECTORY--SEARCH SPEC,LIST FILE?
EO?.* [RETURN]
```

Bemerkung 1: Wenn Sie Dos 2.5 benutzen, um das Inhaltsverzeichnis einer Nicht-DOS Diskette ausgeben zu lassen, kann es vorkommen, daß die Ausgabe keinen Sinn ergibt. Kommerzielle Spiele sind ein Beispiel für solche Disketten. Wenn dies einmal geschehen sollte, dann schalten Sie Ihren Computer aus und starten ihn dann erneut mit der DOS System Diskette in Laufwerk 1.

Bemerkung 2: Sie können nicht den DOS 2.5 DISK DIRECTORY-Befehl benutzen, um das Inhaltsverzeichnis für eine DOS 3 Diskette anzuzeigen. Sehen Sie in Anhang L im COPY32.COM-Kapitel um DOS 3 Dateien zu konvertieren.

B. Run Cartrdige (Zum Modul oder zum Basic)

Jedesmal wenn Sie "B" auswählen, übergibt DOS die Kontrolle Ihres ATARI-Computer an das eingefügte Steckmodul oder an das eingebaute BASIC, wenn vorhanden. Wenn sich das BASIC-Steckmodul im Computer befindet oder wenn das BASIC eingebaut ist, erscheint der Prompt "READY". Wenn sich kein Steckmodul in Ihrem Computer befindet und Sie kein eingebautes BASIC haben, erscheint die Meldung "NO CARTRIDGE" (kein Steckmodul) auf Ihrem Bildschirm.

Warnung: Wenn Sie keine MEM.SAV-Datei auf Ihrer System Diskette hatten wie Sie DOS gestartet haben, werden Sie feststellen, daß jegliches BASIC- oder ASSEMBLER-Programm aus dem internen Speicher gelöscht wurde, bevor Sie in DOS hineinkommen. Sie können Ihr Programm nicht wieder zurückholen, wenn Sie es nicht vorher auf Diskette gespeichert hatten, bevor Sie DOS aufgerufen haben. Dieser Verlust bei der Benutzung von DOS 2.5 ist darauf zurückzuführen, daß Sie sich den Programmspeicher mit dem Disk Utility Package aus der DUP.SYS-Datei teilen müssen. Dieses Teilen hat zur Folge, daß Sie mehr Arbeitsspeicher zu Verfügung haben.

Beispiel:

```
SELECT ITEM OR [RETURN] FOR MENU
B [RETURN]
```

Wenn Sie jedoch die MEM.SAV-Datei auf der Diskette in Laufwerk 1 haben, dann werden die entsprechenden Programme automatisch auf Ihrer Diskette abgespeichert, wenn Sie den Befehl "DOS" eingeben und [RETURN] drücken. Gleichfalls wird das Programm wieder in den Arbeitsspeicher geladen, wenn die Kontrolle wieder an das Steckmodul übergeben wird. Dabei muß die Diskette in Laufwerk 1 dieselbe sein, als Sie DOS aufgerufen haben und daß MEM.SAV nicht entwertet wurde, indem Sie COPY FILE, DUPLICATE FILE oder DUPLICATE DISK benutzt haben. Es wird eine Mitteilung ausgegeben, wenn Sie versuchen einen dieser Befehle auszuführen, um Sie daran zu erinnern, daß MEM.SAV entwertet werden kann (sehen Sie unter N., CREATE MEM.SAV, in diesem Kapitel).

C. Dateien kopieren (Copy File(s))

Wenn Sie zwei oder mehr Diskettenlaufwerke besitzen, und eine Datei von Laufwerk 1 in Laufwerk 2 kopieren wollen, so benutzen Sie diesen Befehl.

Wenn Sie ein XF551- und ein 810-Laufwerk besitzen, und Dateien vom XF551 Laufwerk zum 810-Laufwerk kopieren wollen, müssen Sie zunächst auf dem 810-Laufwerk die Zieldiskette formatieren (oder unter Benutzung der Option P.FORMAT SINGLE auf dem XF551-Laufwerk). Nun können Sie soviele Dateien wie auf der einseitig formatierten Diskette Platz haben kopieren. Es ist nicht möglich Dateien zu kopieren, welche mehr als 707 Sektoren haben. Dies ist die maximale Kapazität einer einseitig formatierten Diskette.

Wenn Sie Disketten haben, welche mit DOS 2.0S bearbeitet, bzw. gespeichert wurden, so sehen Sie bitte im Anhang H für weitere Informationen nach. Haben Sie Disketten, welche mit DOS 3.0 bearbeitet wurden, so sehen Sie im Anhang L nach.

In Verbindung mit dem Befehl COPY FILE gibt es zwei Parameter, FROM und TO. FROM bietet Ihnen die Möglichkeit zu wählen ob Sie Wildcards verwenden wollen, oder nicht. Die Verwendung von Wildcards im ersten Parameter gibt Ihnen die sehr nützliche Möglichkeit bestimmte Gruppen auf einmal von einem Laufwerk zum anderen zu kopieren, siehe Beispiel 6. Die Option /A können Sie mit dem zweiten Parameter nutzen. Der zweite Parameter ist normalerweise ein Pfadname, kann jedoch auch als Ausgabeeinheit benannt werden, so zum Beispiel: P: für Drucker oder D: für Diskettenstation.

COPY FILE kann auch benutzt werden um eine Sicherungskopie bestimmter Dateien auf derselben Diskette abzulegen. Dies geschieht unter demselben Dateinamen mit einem anderen Extender, oder unter einem ganz anderen Dateinamen. Wenn Sie eine Datei kopieren wollen, während sich auf Ihrer Systemdiskette eine MEM.SAVE-Datei befindet, werden Sie eine Prompt-Meldung auf Ihrem Bildschirm bekommen. Diese neue Meldung erhalten Sie nach Eingabe der Kennung des Laufwerkes, von dem die Information kommt, sowie der Kennung des Laufwerks an die die Information gesendet wird. Die Nachricht:

```
TYPE "Y" IF OK TO USE PROGRAM AREA
CAUTION: A "Y" INVALIDATES MEM.SAV
```

soll Sie daran erinnern, daß DOS sich jeden verfügbaren Speicherplatz nehmen kann, um den Kopiervorgang zu beschleunigen. Wenn Sie A mit "Y" bestätigen, sagen Sie damit DOS, daß es sich keine Gedanken über eventuelle MEM.SAVE-Dateien zu machen braucht, diese werden dann einfach gelöscht. Wenn Sie "N" eingeben teilen Sie DOS mit, daß es sich nicht den gesamten Speicherplatz einverleiben darf. DOS nimmt dann nur einen kleinen Speicherbereich, um Ihre Datei zu bewegen. Anders gesagt bedeutet dies, daß er Ihre Datei ebenfalls kopiert, jedoch wesentlich mehr Zeit für den Kopiervorgang benötigt.

Beispiel 1: Kopieren von DOSEX.BAS von Laufwerk 1 zu Laufwerk 2

```
SELECT ITEM OR [RETURN] FOR MENU
C [RETURN]
COPY FROM, TO?
D1:DOSEX.BAS,D2:DOSEX.BAS [RETURN]
```

Beispiel 2: Anfertigen einer Backup-Kopie auf derselben Diskette

```
SELECT ITEM OR [RETURN] FOR MENU
C [RETURN]
COPY FROM, TO?
D1:DOSEX.BAS,D1:DOSEX.BAK
```

Beispiel 3:

```
SELECT ITEM OR RETURN FOR MENU
C [RETURN]
COPY FROM, TO?
D1:DOSEX.LST, E [RETURN]
```

Beispiel 4: Kopieren bestimmter Daten in eine Datei namens TEMP.DAT. Geben Sie die Daten, welche Sie in TEMP.DAT Kopieren wollen, an. CONTROL 3 beendet die Eingabe.

```
SELECT ITEM OR [RETURN] FOR MENU
C [RETURN]
COPY FROM, TO?
E:,D1:TEMP.DAT [RETURN]

PETER, [RETURN]
BILL, [RETURN]
RAY, [RETURN]
STEVE, [RETURN]
[CONTROL] 3
```

Beispiel 5: Sendet die Datei DISEX.LST zum Drucker

```
SELECT ITEM OR [RETURN] FOR MENU
C [RETURN]
COPY FROM, TO?
D1:DISEX.LST,P:
```

Beispiel 6: Kopiert alle Dateien von Laufwerk 1 zu Laufwerk 2 mit Außnahme der Dateien mit dem Extender .SYS

```
SELECT ITEM OR [RETURN] FOR MENU
C [RETURN]
COPY FROM, TO?
*.*,D2: [RETURN]
```

Beispiel 7: Hängt die Datei PROGRAM 1 aus Laufwerk 1 an die Datei PROGRAM 2

```
SELECT ITEM OR [RETURN] FOR MENU
C [RETURN]
COPY FROM, TO?
D1:PROG2,PROG1/A [RETURN]
```

D. Löschen von Dateien (Delete File(s))

Diese Option erlaubt es Ihnen eine oder mehrere Dateien von einer Diskette zu löschen. Diese Dateien werden dann im aktuellen Diskettenverzeichnis nicht mehr dargestellt. Wildcards, auch Joker genannt, können im Pfadnamen verwendet werden.

Bevor Sie mit dem Löschen von Dateien beginnen, haben Sie die Möglichkeit zu bestimmen, ob es vor dem Löschen eine Sicherheitsabfrage gibt, oder nicht. Wählen Sie "N", so wird diese Abfrage einfach ignoriert (siehe dazu Beispiel 3).

Sie können natürlich auch alle Dateien einer Diskette löschen, die Diskette bleibt dann trotzdem formatiert. Das Beispiel 4 zeigt Ihnen die einzelnen Schritte zum Löschen aller Dateien auf Laufwerk 1. Beachten Sie, daß die Sicherheitsabfrage hier ignoriert wird, sodaß DOS nicht bei jeder Datei nachfragt, ob Sie diese Datei wirklich löschen wollen. Wenn Sie versuchen eine speziell gesicherte Datei zu löschen, erhalten Sie von DOS eine Fehlermeldung, auf dem Bildschirm erscheint ERROR-167 (File Locked).

Beispiel 1: Löschen aller Dateien auf Laufwerk 2, welche mit REM beginnen und als Extender .BAS haben. Eine Sicherheitsabfrage für jede Datei.

```
SELECT ITEM OR [RETURN] FOR MENU
D [RETURN]
DELETE FILE SPEC
D2:REM*.BAS RETURN
TYPE "Y" TO DELETE
REM1.BAS?
Y [RETURN]
REMBAA.BAS?
Y [RETURN]
```

Beispiel 2: Löschen einer einzelenen Datei mit Sicherheitsabfrage

```
SELECT ITEM OR [RETURN] FOR MENU
D [RETURN]
DELETE FILE SPEC
D:TEMP.DAT [RETURN]
TYPE "Y" TO DELETE
TEMP.DAT?
"Y" [RETURN]
```

Beispiel 3: Löschen einer Datei ohne Sicherheitsabfrage

```
SELECT ITEM OR [RETURN] FOR MENU
D [RETURN]
DELETE FILE SPEC
DOSEX.BAS/N [RETURN]
```

Beispiel 4: Löschen aller Dateien auf Laufwerk 1

```
SELECT ITEM OR [RETURN] FOR MENU
D [RETURN]
DELETE FILE SPEC
*.*/N [RETURN]
```

E. Umbenennen von Dateien (Rename File)

Diese Option erlaubt es Ihnen, die Namen einer oder mehrerer Dateien zu ändern. Für diese Option gibt es zwei Parameter, OLD NAME und NEW.

OLD NAME ist immer ein kompletter Pfadname, sollten Sie keine Laufwerk-Kennung angeben, geht der Computer von Laufwerk 1 aus. Der Parameter NEW beinhaltet lediglich den neuen Dateinamen. In beiden Parametern können Sie Wildcards benutzen, siehe Beispiel 2. Wann immer Sie Wildcards benutzen, beachten Sie, daß diese Wildcards auch im neuen Dateinamen angegeben werden. Die folgenden Beispiele zeigen Ihnen die Vorgehensweise beim Benutzen von Wildcards.

Korrekte Definitionen:

```
TEST,NEW
TEST.*,NEW.*
*.DAT,*.BAK
*.??1,*.??2
TEST3.DAT,FILE?.*
TEST3.DAT,FILE*.*
```

Unkorrekte Definitionen:

```
TEST.*,NEW
TEST?,NEW
TEST*.??1,TEST3.??2
```

Merken Sie sich, daß Sie im neuen Dateinamen mehr Wildcards als im alten verwenden können, in diesem Fall werden die Charaktere des alten Dateinamens mit übernommen. Dies wird Ihnen im letzten korrekten Beispiel gezeigt, die Datei TEST3.DAT wird als FILE3.DAT abgespeichert.

Sie erinnern sich sicher, daß jede Datei einen unverwechselbaren Dateinamen hat. Wenn Sie nun einzelne Dateien umbenennen ohne Wildcards zu verwenden, so können Sie unter DOS 2.5 der Datei jeden erdenklichen Namen geben, auch wenn sich auf Ihrer Diskette bereits eine Datei mit dem gleichen Namen befindet. Wollen Sie eine dieser Dateien nun weiter bearbeiten, zum Beispiel Löschen, so greift DOS auf beide Dateien zu. Wenn Sie merken, daß sich auf Ihrer Diskette zwei Dateien mit gleichem Namen befinden, so haben Sie die Möglichkeit diese wieder umzubenennen, DOS 2.5 ändert dann den Namen der zuerst gefundenen Datei in den Namen , welchen Sie zuvor spezifiziert haben.(Sehen Sie dazu auch im Kapitel DISKFIX.COM ,Rename File # Anhang L nach).

Wenn Sie versuchen auf einer schreibgeschützten Diskette einen Dateinamen zu ändern, so erhalten Sie auf dem Bilddschirm die Meldung ERROR-144. Wenn Sie versuchen den Namen einer Datei zu ändern, welche sich nicht auf der von Ihnen angegebenen Diskette befindet, so erhalten die die Fehlermeldung ERROR-170 (File not found). Erhalten Sie als Fehlermeldung ERROR-167, so haben Sie versucht eine geschützte Datei umzubenenen.

Beispiel 1: Ändern der Datei TEMP.DAT in Laufwerk 1 in NAMES.DAT

```
SELECT ITEM OR [RETURN] FOR MENU
E [RETURN]
RENAME-GIVE OLD NAME , NEW
D2:TEMP.DAT, NAMES.DAT [RETURN]
```

Beispiel 2: Alle Dateien in Laufwerk 1, welche den Extender 8KB haben, sollen den Extender .BAS bekommen

```
SELECT ITEM OR [RETURN] FOR MENU
E [RETURN]
RENAME- GIVE OLD NAME, NEW
*.8KB, *.BAS [RETURN]
```

F. Sichern von Dateien (Lock File)

Nutzen Sie diese Funktion um einzelne Dateien vor versehentlichen Überschreiben zu schützen. Derart geschützte Dateien können nicht gelöscht, überschrieben oder umbenannt werden. Sollten Sie versuchen eine geschützte Datei zu überschreiben, so erhalten Sie als Fehlermeldung ERROR-167. Sie können auch hier Wildcards benutzen, um mehrere Dateien auf einmal zu schützen. Eine geschützte Datei wird im Disketten-Inhaltsverzeichnis mit einem Stern gekennzeichnet. Verwechseln Sie diesen Stern jedoch nicht mit dem Wildcard-Symbol.

Warnung: Haben Sie mehrere Dateien auf einer Diskette geschützt, und formatieren nun diese Diskette, so werden auch die geschützten Dateien gelöscht.

Beispiel 1: Schützen der Datei DOS.SYS im Laufwerk 1

```
SELECT ITEM OR [RETURN] FOR MENU
F [RETURN]
WHAT FILE TO LOCK?
DOS.SYS [RETURN]
```

Beispiel 2: schützt alle Dateien in Laufwerk 1, welche als Extender .BAS haben

```
SELECT ITEM OR [RETURN] FOR MENU
F [RETURN]
WHAT FILE TO LOCK?
D1:*.BAS [RETURN]
```

Beispiel 3: Schützt alle Dateien in Laufwerk 1, welche mit T beginnen

```
SELECT ITEM OR [RETURN] FOR MENU
F [RETURN]
WHAT FILE TO LOCK?
T*.* [RETURN]
```

Beispiel 4: Schützt alle Dateien in Laufwerk 1

```
SELECT ITEM OR [RETURN] FOR MENU
F [RETURN]
WHAT FILE TO LOCK?
*.* [RETURN]
```

G. Entsichern von Dateien (Unlock File)

Mit Hilfe dieser Funktion können Sie gesicherte Dateien wieder entsichern. Wenn Sie Ihre gesicherten Dateien wieder entsichert haben, verschwindet der Stern hinter dem Dateinamen, sobald Sie das Disketten-Inhaltsverzeichnis aufrufen. Sie können auch hier wieder Wildcards in den Pfadnamen verwenden.

Beispiel 1: Entsichert DOSEX.BAS in Laufwerk 1

```
SELECT ITEM OR [RETURN] FOR MENU
G [RETURN]
WHAT FILE TO UNLOCK?
DOSEX.BAS [RETURN]
```

Beispiel 2: Entsichert alle Dateien in Laufwerk 1, welche mit T beginnen

```
SELECT ITEM OR [RETURN] FOR MENU
G [RETURN]
WHAT FILE TO UNLOCK?
T*.* [RETURN]
```

Beispiel 3: Entsichert alle Dateien in Laufwerk 1, welche mit PROB beginnen, und mit .DAT enden.

```
SELECT ITEM OR [RETURN] FOR MENU
G [RETURN]
WHAT FILE TO UNLOCK?
PROB?.DAT [RETURN]
```

H. Schreiben von DOS-Dateien (Write DOS Files)

Um DOS 2.5 - Dateien auf eine Diskette schreiben zu können, muß diese Diskette vorher unter Benutzung von DOS 2.5 oder DOS 2.0s formatiert worden sein (siehe auch Format Disk).

Sofort nachdem DOS-Dateien auf die Diskette geschrieben wurden, erscheint wieder das Haupt-Menu, sowie der Prompt SELECT ITEM OR RETURN FOR MENU.

Wenn Sie nun eine neue DOS-Datei auf Ihre Diskette schreiben wollen, diese jedoch schreibgeschützt ist, so erhalten Sie die Fehlermeldung ERROR-144.

```
SELECT ITEM OR [RETURN] FOR MENU
H [RETURN]
DRIVE TO WRITE DOS FILES TO?
1 [RETURN]
TYPE "y" TO WRITE DOS TO DRIVE 1
Y [RETURN]
WRITING NEW DOS FILES
```

I. Format Disk

Diese Option wird benutzt, um eine Diskette im erweiterten Format zu formatieren, vorausgesetzt Sie benutzen ein XF551-Laufwerk. Immer wenn Sie diese Option aktivieren und DOS merkt, daß Sie ein 810-Laufwerk benutzen wird die eingelegte Diskette in normaler Dichte formatiert, da mit einem 810-Laufwerk Informationen nur mit einfacher Dichte gespeichert und bearbeitet werden können.

Die Option P. aus dem DOS 2.5 - Menu. FORMAT SINGLE kann benutzt werden, um den Formatierungsprozess von Disketten mit einfacher Dichte auf einem XF551-Laufwerk zu beschleunigen.

Die zu formatierende Diskette kann leer, aber auch mit Informationen, welche Sie nicht mehr benötigen belegt sein. Das Formatieren einer Diskette schreibt Informationen auf die Diskette, welche es ihr erlauben Daten aufzunehmen, beziehungsweise wiederzugeben.

Eine Diskette welche mit dieser Option und einem XF551-Laufwerk formatiert wurde, kann Informationen in 1023 Sektoren speichern, wogegen eine Diskette mit einfacher Dichte nur über 719 Sektoren verfügt. Beim Formatieren, egal in welcher Dichte, wird immer ein Teil Ihrer Diskette für DOS reserviert. Dies erklärt auch warum eine frische in einfacher Dichte formatierte Diskette beim Anschauen des Disketten-Inhaltsverzeichnisses "nur" 707 freie Sektoren anzeigt, in doppelter Dichte sind es 999 freie Sektoren. Beim Formatieren mit doppelter Dichte stehen Ihnen eigentlich aktuell 1.010 freie Sektoren zur Verfügung, jedoch benötigt DOS 2.5 eine dreistellige Nummer um eine Kompatibilität zwischen DOS 2.5 und DOS 2.0S herzustellen.

Wenn Ihr Computersystem sowohl ein XF551-Laufwerk als auch ein 810-Laufwerk beinhaltet, oder wenn Sie mit DOS 2.0S erstellte Disketten bearbeiten schauen Sie in Anhang H für weitere Informationen nach.

Wenn Sie Disketten benutzen wollen, welche mit DOS 3.0 erstellt wurden, schauen Sie unter Anhang L nach. Vergessen Sie nicht auf Ihren Disketten-Labels zu vermerken, mit welchem DOS Sie diese Diskette bearbeitet haben.

Das nachfolgende Beispiel geht davon aus, daß Sie die Diskette in Laufwerk 1 formatieren wollen, natürlich können Sie auch Disketten in Laufwerk 2 formatieren. Wenn Sie eine Diskette formatieren wollen, und diese Diskette schadhafte Sektoren hat - was leider immer mal wieder vorkommen kann - so erscheint die Fehlermeldung ERROR-173 auf Ihrem Bildschirm. Wenn DOS die Meldung bekommt das eine Diskette schadhafte Sektoren hat, so versucht es trotzdem diese Diskette zu formatieren. Wenn dies passiert, kann es bis zu 15 Minuten dauern, bis ERROR-173 wieder auftaucht.

Wenn Sie einmal eine neue schadhafte Diskette erwischt haben, so gehen Sie unverzüglich zu Ihrem Fachhändler, und tauschen sie um.

SELECT ITEM OR [RETURN] FOR MENU
I [RETURN]
WHICH DRIVE TO FORMAT?
1 [RETURN]
TYPE "Y" TO FORMAT DISK 1
Y [RETURN]

Warnung: Das Formatieren einer Diskette löscht alle Daten auf dieser Diskette.

J. Anfertigen einer Disketten-Kopie (Duplicate Disk)

Benutzen Sie diese Option um sich eine exakte Kopie Ihrer Originaldiskette anzufertigen. Sie können diese Option benutzen, um auf einem Laufwerk eine Kopie anzufertigen, wobei Sie mehrere Male die Originaldiskette gegen die Zieldiskette auswechseln müssen, solange bis der Kopiervorgang beendet ist. Wenn Sie über mehrere Laufwerke verfügen, so legen Sie die Originaldiskette in Laufwerk 1 und die Zieldiskette in Laufwerk 2. DOS 2.5 erledigt den Kopiervorgang dann automatisch. Bitte beachten Sie auch hier, daß Sie eine mit doppelter Dichte formatierte Diskette nicht als Zieldiskette in einem 810-Laufwerk verwenden können. Wenn Sie über ein XF551 - und ein 810-Laufwerk verfügen, so schauen Sie bitte in Anhang H für weitere Informationen nach. Für DOS 3.0 schauen Sie unter Anhang L nach.

Der Kopiervorgang ist eine "Sektor-für-Sektor"-Technik. Dies bedeutet, daß Ihre Dateien nicht nur kopiert werden, sondern auf der Zieldiskette auch im selben Sektor stehen wie auf der Originaldiskette. Das Inhaltsverzeichnis Ihrer Originaldiskette wird ebenfalls auf die Zieldiskette mitkopiert. Dies ist auch einer der Gründe warum Dateien welche sich vorher auf Ihrer Zieldiskette befanden gelöscht werden. Deshalb passen Sie auf, daß sich auf Ihrer Zieldiskette keine wichtigen Daten mehr befinden.

Bevor Sie anfangen eine Kopie Ihrer Originaldiskette anzufertigen, sollten Sie ebenfalls die sich im RAM befindliche Sprache, zum Beispiel BASIC auf einer Diskette sichern, da während des Kopiervorgangs jeder verfügbare Speicher benutzt wird. Dies unterscheidet die Option DUPLICATE DISK im wesentlichen von der Option COPY DISK. Bei der Option DUPLICATE DISK wird der rechner-interne Speicher benötigt, um die Dateien von der Originaldiskette zwischenzuspeichern, bevor diese auf die Zieldiskette geschrieben werden.

Anfertigen einer Diskettenkopie mit einem Laufwerk

Beim Anfertigen einer Disketten-Kopie mit einem Laufwerk, ist dieses Laufwerk sowohl Original- als auch Ziel-Laufwerk, siehe nachfolgendes Beispiel.

Schützen Sie Ihre Originaldiskette mit dem Schreibschutz. Sollten Sie diese Diskette nun als Zieldiskette verwenden wollen, so erscheint auf dem Bildschirm die Fehlermeldung ERROR-144, und Ihre Originaldiskette bleibt erhalten. Wenn Sie nun einen anderen Befehl als "Y" eingeben und mit RETURN bestätigen, jedoch "Y" als Antwort für IF OK TO USE PROGRAM AREA erwartet wird, so bricht das Programm ab und kehrt zum Hauptmenu zurück. Der Prompt SELECT ITEM OR RETURN FOR MENU erscheint.

Hier nun das Beispiel für das Anfertigen einer Disketten-Kopie mit einem einzelnen Laufwerk:

```
SELECT ITEM OR [RETURN] FOR MENU
J [RETURN]
DUP DISK SOURCE, DEST DRIVES
1,1 [RETURN]
INSERT SOURCE DISK, TYPE RETURN
[RETURN]
INSERT DESTINATION DISK, TYPE RETURN
[RETURN]
```

Anmerkung: Die Länge des Kopiervorgangs richtet sich danach, wie groß die Dateien auf Ihrer Originaldiskette sind, beziehungsweise wieviele Dateien auf einmal in den rechner-internen Speicher passen.

Anfertigen einer Diskettenkopie mit mehreren Laufwerken

Mit mehreren Laufwerken ist das Anfertigen einer Disketten-Kopie relativ einfach. Bitte bedenken Sie jedoch wieder, daß Sie in doppelter Dichte formatierte Disketten nicht in einem 810-Laufwerk nutzen können. Eine Beschriftung Ihrer Disketten hilft Ihnen diesen Fehler zu vermeiden. Auch bei mehreren Laufwerken gilt, daß Sie vorher alle Programme aus dem rechner-internen Speicher nehmen, da sie sonst unweigerlich gelöscht werden. Beim Kopieren mit mehreren Laufwerken ist das Laufwerk 1 das Original-Laufwerk, Laufwerk 2 gilt als Ziel-Laufwerk. Das Kopieren geht hier wesentlich schneller vonstatten als mit einem Laufwerk, aber auch hier hängt die Zeit davon ab, wie voll Ihre Originaldiskette ist.

```
SELECT ITEM OR [RETURN] FOR MENU
J [RETURN]
DUP DISK SOURCE, DEST DRIVES
1,2 [RETURN]
INSERT BOTH DISKS, TYPE RETURN
[RETURN]
```

K. BINÄRES SICHERN (Binary Save)

Diese Option wird sicherlich nicht von einem Programmier-Anfänger benutzt. Wenn Sie noch keine Erfahrung im Hexa-Dezimal-System und der Assembler-Programmiersprache haben, so werden Sie kaum in der Lage sein die Bedeutung der Zahlen im folgenden Beispiel zu verstehen.

Benutzen Sie diese Option um die Inhalte der Speicherbereiche als Objekt-Code zu sichern. Programme, welche mit dem Assembler-Editor-Steckeinsatz geschrieben wurden, haben das gleiche Format. Die Parameter für diese Option -START,END,INIT,RUN - sind als hexadezimale Ziffern angegeben. Die Start- und Endadresse sind für jede Datei und jedes Programm gleich. Die Parameter INIT und RUN sind optionale Adressen, welche es Ihnen erlauben direkt an bestimmte Speicherplätze zu springen, und die dort vorhandenen Informationen zu lesen.

Im nachfolgenden Beispiel wird eine Datei namens BINFIL.OBJ aufgerufen, welche auf der in Laufwerk 1 befindlichen Diskette die Startadresse 3C00 und als Endadresse 5BFF hat.

Beispiel 1:

```
SELECT ITEM OR [RETURN] FOR MENU
K [RETURN]
SAVE-GIVE FILE, START,END(,INIT,RUN)
BINFIL.OBJ,3C00, 5BFF [RETURN]
```

Weitergehende Informationen über optionale Parameter

Alle binären Dateien haben als Programmkopf eine feste sechs Byte große Adresse, welche die Datei charakterisiert. Aus diesen Daten können Sie sich ganz leicht die Werte der START- und END-Adresse des oben beschriebenen Beispiels heraussuchen.

Die zwei optionalen Parameter INIT und RUN sorgen dafür, daß ein in Assembler geschriebenes Programm sofort nach Aufruf lauffähig ist. Solche Programme nennt man auch "load-and-go"-Dateien. Eine Datei welche keine Daten für diese Optionen enthält nennt man "load"-Datei, da solche Dateien nur geladen werden, sich jedoch nicht selber starten, bevor Sie den Befehl RUN AT ADRESS eingeben.

Programmkopf Byte	Dezimal nummer	HEX nummer	Beschreibung
# 1	255	FF	Identifizierungs-Code für
# 1	·255	FF	binary load File
# 2	0	00	Startadresse (LSB)
# 4	60	3C	(MSB)
# 5	255	FF	Endadresse (LSB)
# 6	91	5B	(MSB)

Im allgemeinen definiert der RUN-ADRESS-Parameter den Punkt im Programm, wo die Ausführung des Programms beginnt, sobald das Programm in den rechnerinternen Speicher geladen wurde. Aus diesem Grund kann es immer nur eine definitive RUN-Adresse geben, vorausgesetzt das Programm ist zusammenhängend. Sollte das Programm aus mehreren kleinen Programmen bestehen, so kann jedes einzelne dieser kleinen Programme eine eigene RUN-Adresse haben. In diesem Fall würde das Programm erst starten, nachdem die letzte RUN-Adresse eingelesen wurde.

Wenn eine INIT-Adresse spezifiziert ist und die aktuelle Adresse in den rechnerinternen Speicher geladen wurde, wird die INIT-Adresse gelöscht. Das passiert immer dann, wenn eine Datei aus mehreren aneinanderhängenden "load-and-go"-Dateien besteht. In diesem Fall wird jede INIT-Adresse wenn Sie geladen wurde, gelöscht.

Die RUN-Adresse ist immer im Bereich 2E0(LOW) und 2E1(HIGH). Die INIT-Adresse ist immer im Bereich 2E2(LOW) und 2E3(HIGH). Bedenken Sie, daß die INIT-Adresse immer sofort nach dem Laden gelöscht wird.

Binäres Sichern mit optionalen Parametern

Das nachfolgende Beispiel zeigt Ihnen ein in Assembler geschriebenes Programm, welches einen Datenbereich benötigt, der zuvor erst einmal initialisiert werden muß, bevor das Hauptprogramm diesen Bereich nutzen kann.

Nehmen wir einmal an der Initialisierungs-Code belegt einen Platz von 4000(HEX) bis 41FF(HEX), und das Hauptprogramm belegt einen Platz von 4200(HEX) bis 4FF(HEX). Nehmen wir ferner an das sowohl der Initialisierungs-Code als auch das Hauptprogramm Code beinhalten welcher gelöscht wird, und das der Initialisierungs-Code mit RTS (Return) endet.

Im folgenden Beispiel wird angenommen, daß sich das Programm LAGPRG.OBJ im Speicher befindet.

Beispiel:

```
SELECT ITEM OR [RETURN] FOR MENU
K [RETURN]
SAVE-GIVE FILE,START,END(,INIT,RUN)
LAGPRG.OBJ,4000,4FF,4000,4200 [RETURN]
```

Die folgenden Ereignisse geschehen, wenn diese Datei in den Speicher geladen wird:

1. Der Speicherbereich von 4000 bis 4FF wird mit dem Programm gefüllt.

2. Die INIT-Adresse 4000(HEX) wird im Speicherbereich 2E2 und 2E3 gespeichert.

3. Das Initialisierungsprogramm, welches den Speicherbereich von 4000 bis 41FF belegt, wird gelöscht.

4. Die RUN-Adresse 4200(HEX) wird im Speicherbereich 2E0 und 2E1 gespeichert.

5. Das Hauptprogramm im Speicherbereich 4200 bis 4FFF beginnt zu laufen, solange bis ein Return,Reset oder Break diesen Vorgang abbricht.

Struktur einer zusammengesetzten binären Datei

Bevor Sie sich das nächste Beispiel ansehen, werfen Sie einen Blick auf die Struktur einer zusammengesetzten binären Datei.

Eine zusammengesetzte Datei besteht aus mehreren einzelnen und unterschiedlichen Dateien, welche immer aneinander angehängt wurden. Sie können solche Dateien auf zwei verschiedenen Wegen kreieren. Der eine Weg ist der, die Option COPY FILE in Verbindung mit der Attend-Option zu nutzen. Dateien, welche auf diese Art und Weise geschaffen wurden, sind nicht kompatibel zum Assembler-Editor-Loader, sie können nur über die Option BINARY LOAD des DOS geladen werden.

Wenn BINARY SAVE benutzt wird, sind die einzelnen Identifikations-Codes, außer dem ersten auf der Kopie nicht mehr vorhanden. Dies ist die einzige Dateiform, welche mit dem load-Kommando des Assembler-Editor-Steckmoduls kompatibel ist. Die BINARY LOAD-Option des DOS ist dagegen mit beiden Dateiformen kompatibel.

Nun überlegen Sie mal, was passiert wenn eine Datei wie diese geladen wird, beachten Sie die verschiedenen INIT- und RUN-Adressen welche für jedes einzelne Segment Ihrer Datei spezifiziert wurden, bevor diese Segmente zusammengezogen wurden. Dies hilft Ihnen die INIT- und RUN-Adressen als Teil Ihrer Daten in jedem Segment zu sehen, was Sie im Grunde genommen auch sind.

Beispiel:
Stellen Sie sich vor, Sie haben drei Dateien, jede davon hat eine RUN-Adresse, es sind jedoch keine INIT-Adressen in den Daten vorhanden. Dieses Beispiel zeigt Ihnen einen Weg, um solch eine Datei zu erstellen.

```
SELECT ITEM OR [RETURN] FOR MENU
K [RETURN]
SAVE-GIVE FILE, START, END (,INIT, RUN)
PART1.OBJ, 2000, 21FF,, 2000 [RETURN]

SELECT ITEM OR [RETURN] FOR MENU
K [RETURN]
SAVE-GIVE FILE, START, END (,INIT, RUN)
PART2.OBJ, 2200, 23FF,, 2200 [RETURN]
```

Die anderen beiden Dateien, PART2.OBJ und PART3.OBJ, welche genau wie PART1.OBJ erstellt werden, können nun in die Datei WHOLE.OBJ eingegliedert werden. Dazu benutzen Sie die Optionen BINARY SAVE oder COPY FILE mit Attend-Option. Wenn diese neue Datei nun geladen wird, passiert folgendes:

1. PART1.OBJ wird geladen, das Programm wird jedoch nicht gestartet, da es kein INIT gibt.

2. Die RUN-Adresse für PART1.OBJ wird im Bereich 2E0 und 2E1 gespeichert.

3. PART2.OBJ wird geladen, das Programm wird jedoch widerum nicht gestartet, da ein INIT fehlt.

4. Die RUN-Adresse für PART2.OBJ wird im Speicherbereich 2E0 und 2E1 gespeichert, die RUN-Adresse von PART1.OBJ wird überschrieben.

5. PART3.OBJ wird geladen, das Programm wird nicht gestartet.

6. Die RUN-Adresse für PART3.OBJ wird im Speicherbereich 2E0 und 2E1 gespeichert, die RUN-Adresse von PART2.OBJ wird überschrieben.

7. Die Ausführung des Programms beginnt bei der RUN-Adresse von PART3.OBJ, wenn Sie nun am Ende Ihrer Datei angelangt sind.

Beispiel:
Ein anderes Beispiel für eine zusammengesetzte Datei, die Datei BIGFILE.OBJ ist aus drei verschiedenen Dateien zusammengesetzt. Bedenken Sie, daß jede Datei einen anderen Teil Ihres Speichers belegt und das -SEG1.OBJ zwar eine INIT-Adresse, jedoch keine RUN-Adresse hat

SEG2.OBJ weder eine INIT- noch eine RUN-Adresse hat
SEG3.OBJ eine INIT-Adreese hat, und auch eine RUN-Adreese für
SEG2.OBJ und am Anfang von SEG1.OBJ angefügt wird.

Wenn nun BIGFILE.OBJ geladen wird, passiert folgendes:

1. SEG1.OBJ wird geladen

2. SEG1.OBJ wird ab seiner INIT-Adresse gestartet

3. SEG2.OBJ wird geladen

4. SEG3.OBJ wird an den Anfang der Datei SEG1.OBJ geladen

5. SEG3.OBJ startet an seiner INIT-Adresse

6. SEG2.OBJ startet an der RUN-Adresse, welche in SEG3.OBJ definiert wurde

Wie Sie sehen, können Sie mit dieser Option das Maximale an Geschwindigkeit und Flexibilität bei großen Dateien erreichen, welche sofort nach dem Laden selbstständig anfangen zu arbeiten.

Beispiel:
Um eine bestehende "load-only"-Datei in eine "load-and-go"-Datei zu konvertieren, laden Sie diese Datei in den rechner-internen Speicher und speichern ihn dann mit einem neuen Dateinamen ab. Dazu benutzen Sie bitte die Option BINARY SAVE. Dies birgt jedoch auch noch einige Probleme, wenn Sie die Schlußadresse Ihrer Datei vergessen haben, beziehungsweise wenn die einzelnen Segmente Ihrer Datei nicht aufeinanderfolgend im Speicher abgelegt sind. Wenn dies passiert ist, kann es sein, daß die konvertierte Datei mehr Speicherplatz auf Ihrer Diskette belegt wie zuvor.

Sie können diese Probleme vermeiden, wenn Sie die nachfolgend beschriebene Prozedur verwenden. Dieses Beispiel zeigt Ihnen das Einladen einer "load"-Datei mit der Startadresse 4000 HEX, welche in eine "load-and-go"-Datei umgewandelt werden soll. In diesem Beispiel soll eine Ein-Byte-Datei mit der Speicheradresse FF00 an das Ende der Datei LOADFIL.OBJ angehängt werden. Wenn die Startadresse dieser Datei an der gleichen Stelle liegt an welcher normalerweise unsere "load"-Datei zu arbeiten beginnt, so soll nach dem vollständigen Einladen der Datei mit der Ausführung des Programmes begonnen werden.

```
SELECT ITEM OR [RETURN] FOR MENU
K [RETURN]
SAVE-GIVE FILE, START, END (,INIT, RUN)
LOADFIL.OBJ /A, FF00, FF00,, 4000
```

L. Laden einer Binär-Datei (Binary Load)

Benutzen Sie die Funktion BINARY LOAD um eine in Assembler geschriebene Datei in den rechner-internen Speicher zu laden. Diese Datei muß entweder mit Hilfe der Option K gesichert worden sein, oder aber mit dem Assembler-Editor-Steckmodul erstellt worden sein. Wenn die RUN- oder INIT-Adresse an die Datei mit dem Speicherplatz 2E0 und 2E1 oder 2E2 und 2E3 angehängt wird so soll direkt nach dem Ende des Ladevorgangs das Programm automatisch gestartet werden. Die INIT- und RUN-Adresse werden ignoriert, wenn Sie nach dem Dateinamen /N eingeben, siehe Beispiel 1. Das Programm kann dann über die Option RUN AT ADRESS gestartet werden.

Beispiel 1:

```
SELECT ITEM OR [RETURN] FOR MENU
L [RETURN]
LOAD FROM WHAT FILE?
MYFILE.OBJ /N [RETURN]
```

Das Benutzen dieser Funktion ohne /N wird im zweiten Beispiel gezeigt. Wenn diese Datei die Startadresse 2E0 hat und 2E1 ist an diese angehängt, so beginnt diese Datei mit der Ausführung des Programmes sobald die Datei vollständig eingelesen wurde.

Beispiel 2:

```
SELECT ITEM OR [RETURN] FOR MENU
L [RETURN]
LOAD FROM WHAT FILE?
BINFIL.OBJ [RETURN]
```

Das Beispiel 3 zeigt Ihnen eine Datei namens MACHL.OBJ, welche weder eine Start- noch eine INIT-Adresse hat. In diesem Fall wird nach dem Einladen dieser Datei der Prompt "SELECT ITEM OR RETURN FOR MENU" auf Ihrem Bildschirm erscheinen.

Beispiel 3:

```
SELECT ITEM OR [RETURN] FOR MENU
L [RETURN]
LOAD FROM WHAT FILE?
MACHL.OBJ [RETURN]
```

M. Starten an einer hexadezimalen Adresse (Run at address)

Diese Funktion können Sie benutzen, um eine hexadezimale Startadresse für eine Objekt-Datei einzugeben, nachdem Sie diese mittels der Option BINARY LOAD in den Speicher geladen haben. Diese Funktion wird benutzt wenn keine Startadresse an das Ende dieser Datei angehängt wurde.

In dem folgenden Beispiel wird an der mit 3000 bezeichneten Speicheradresse das Programm gestartet. Seien Sie bei der Eingabe dieser Startadressen sehr vorsichtig, da es Probleme geben kann wenn an dieser Speicherstelle kein ausführbares Programm steht. Zum Beispiel kann es passieren, das Ihnen Ihr Rechner abstürzt und Ihre Daten für immer verloren sind.

SELECT ITEM OR RETURN FOR MENU
M RETURN
RUN FROM WHAT ADRESS?
3000 RETURN

N. Anfertigen einer MEM.SAV-Datei

Diese Funktion erlaubt es Ihnen eine Datei einzurichten, in welche die sich im rechner-internen Speicher befindlichen Daten gesichert werden, sobald Sie DOS aufrufen. Wenn Sie DOS und anschließend RETURN eingeben, sichert der Computer alle im Speicher befindlichen Daten in diese Datei (inklusive des RAM-residenten Benutzerprogrammes, falls Sie eines benutzen), bevor er die Datei DUP.SYS in den Speicher lädt. Wenn Sie nun die Arbeit mit den DOS-Optionen beendet haben, so geben Sie nun B gefolgt von Return ein, um wieder mit Ihrem Steckmodul arbeiten zu können. Alle in der Datei MEM.SAVE gesicherten Daten werden nun wieder in den rechner-internen Speicher geladen. Wenn Sie kein Steckmodul verwenden, so hat das Kommando B keine Wirkung. Auf Ihrem Bildschirm erscheint nun wieder der Prompt "SELECT ITEM OR RETURN FOR MENU".

Bitte beachten Sie, daß Sie DOS nicht sämtlichen zur Verfügung stehenden Speicherplatz geben dürfen, wenn Sie mit den Optionen COPY FILE, DUPLICATE FILE oder DUPLICATE DISK Ihre Daten sichern wollen. DOS weiß nämlich nicht, ob alle, oder nur ein Teil Ihrer Daten in der Datei MEM.SAVE gesichert wurden. Wenn DOS sämtlichen Speicherplatz zur Verfügung gestellt bekommt, so löscht er automatisch die Datei MEM.SAVE . Wenn dies passiert, kann Ihre Datei nicht zurückgeladen werden, wenn Sie die Kontrolle wieder an das Steckmodul übergeben. Hier nun die einzelnen Schritte um auf einer in Laufwerk 1 befindlichen Diskette eine MEM.SAVE-Datei anzulegen. Beachten Sie das eine MEM.SAVE-Datei nur auf Laufwerk 1 angelegt werden kann.

```
SELECT ITEM OR [RETURN] FOR MENU
N [RETURN]
TYPE "Y" TO CREATE MEM.SAV
Y [RETURN]
```

Wenn sich auf dieser Diskette bereits eine MEM.SAV-Datei befindet, so erscheint auf dem Bildschirm die Meldung "MEM.SAV FILE ALREADY EXISTS" und Sie kehren zum Hauptmenu zurück.

Warum sollte man eine MEM.SAV Datei haben?

Diese spezielle Datei erlaubt es Ihnen, Ihr speicherresidentes Programm in eine Datei auf Diskette zu speichern. Der Effektivität willens, muß diese Datei sich auf einer in Laufwerk 1 befindlichen Diskette befinden. Diese Diskette darf nicht schreibgeschützt sein, wenn mit MEM.SAV gearbeitet wird. Wenn MEM.SAV auf Ihrer Diskette existiert und nun DOS aufgerufen wird, so wird die sich im Speicher befindliche Datei DUP.SYS in dieser Datei gesichert. Wenn Sie mit einem Basic-Programm arbeiten, nun aber kurz wieder mit DOS arbeiten wollen, so können Sie MEM.SAV benutzen, ohne vorher Ihre Datei auf Diskette gesichert zu haben. Wenn Sie die Arbeiten mit DOS abgeschlossen haben, und die Kontrolle des Systems wieder an BASIC übergeben, werden automatisch wieder die in MEM.SAV gesicherten Daten in den Speicher gelesen und Ihr Basic-Programm erscheint wieder auf Ihrem Bildschirm.

Hier nun ein Beispiel zum Nutzen der MEM.SAV-Datei:

1. Type LOAD "D:MYPROG.BAS" Return

2. Bearbeiten Sie nun Ihr Programm und beenden Sie die Arbeiten mit der Eingabe RUN gefolgt von Return. Ihr Programm läuft nun und Sie wollen es umbenennen um sich davon eine Backup-Kopie auf Diskette zu sichern.

3. Type SAVE "D:MYPROG.NEW" Return

4. Type DOS return

5. Danach geben Sie ein E für Umbenennen einer Datei ein gefolgt von MYPROG.BAS, MYPROG.OLD Return

6. Geben Sie wieder E ein und drücken Sie Return. Danach geben Sie nun MYPROG.NEW, MYPROG.BAS ein und bestätigen dies wiederum mit Return.

7. Um nun zu Basic zurückzukehren geben Sie B ein.

Benutzen von MEM.SAV um Assembler-Dateien zu schreiben

Die Datei MEM.SAV erlaubt Ihnen, das Schreiben von Assembler-Programmen, die sich den rechner-internen Speicher mit DUP.SYS teilen. Dies bedeutet das Sie Daten aus dem Bereich von LOMEM in den Bereich von HIMEM laden können.

Beispiel:
Nehmen wir einmal an, Sie hätten eine Programm-Datei welche gleich nach dem Laden das Programm ausführen soll. Solche Dateien nennt man "load-and-go-Dateien". In diesen Dateien ist die Startadresse bereits programmiert, sodaß Sie, um das Programm zu starten, nicht mehr die Option RUN AT ADRESS aktivieren müssen. In diesem Fall ist es nicht notwending, daß Sie eine MEM.SAV-Datei auf Ihrer Diskette haben. Die einfachste Möglichkeit dann zu DOS zurückzukehren, ist es Ihren Computer neu zu booten. Wenn Ihre Datei DUP.SYS nicht überschrieben wurde, so können Sie nun durch Eingeben von RETURN direkt wieder in Ihr Programm gelangen. Wenn nun eine Datei Ihre DUP.SYS überschrieben hat, während Sie geladen wurde, merkt dies DOS und lädt automatisch DUP.SYS nachdem das RETURN getätigt wurde.

Benutzen von MEM.SAV um binäre Dateien zu laden

Dieses Kapitel geht von einer binären Datei aus, welche nach dem Laden nicht automatisch gestartet wird, oder Daten für ein anderes Programm beinhaltet. Wenn Ihre einzuladende Datei nicht Teile von DUP.SYS überschreibt, so brauchen Sie keine MEM.SAV-Datei auf Ihrer Diskette. Sollte diese Datei jedoch Teile von DUP.SYS überschreiben, so sollten Sie eine MEM.SAV-Datei auf Ihrer Diskette in Laufwerk 1 eingerichtet haben. Wenn Sie eine MEM.SAV-Datei eingerichtet haben passiert nach dem Anwählen von LOAD BINARY FILE folgendes:

1. Sie benutzen die LOAD BINARY FILE Auswahl um Ihre Datei auszuwählen.

2. Ihre Originaldatei MEM.SAV wird von der Diskette in den Arbeitsspeicher geladen, DUP.SYS wird überschrieben und gelöscht.

3. Ihre Datei wird in die Original MEM.SAV-Datei geladen, und überschreibt dabei den Inhalt der MEM.SAV-Datei.

4. Ihre neue MEM.SAV-Datei wird nun auf dem MEM.SAV-Bereich der Diskette gesichert.

5. DUP.SYS wird von der Diskette in den Arbeitsspeicher zurück geladen.

6. Sie kehren zu DOS zurück, wenn Sie wählen:

RUN CARTRIDGE
Ihre Datei wird in den Arbeitsspeicher geladen, die Kontrolle übernimmt Ihr Steckmodul.

RUN AT ADRESS
Ihre Datei wird geladen und beginnt mit der Ausführung des Programms.

LOAD BINARY FILE
wenn Sie eine "load-and-go"-Datei laden möchten. Wenn die neue Datei einen Teil des Speicherplatzes von DUP.SYS belegt, jedoch nicht das Original, so wird neben Ihrer Datei auch gleichzeitig MEM.SAV in den Hauptspeicher geladen. Wenn Ihre Datei keinen Speicherplatz von DUP.SYS belegt, so wird nur die Datei geladen.

Anmerkung: Wenn Sie zwei Dateien gleichzeitig im Speicher halten wollen, wobei die eine im Speicherbereich von DUP.SYS und die andere außerhalb dieses Speicherbereiches sein soll, so fassen Sie zwei Dateien zu einer Datei zusammen, und laden diese neue Datei in den Arbeitsspeicher.

O. Vervielfältigen von Dateien (Duplicate File)

Diese Option (in Kapitel I beschrieben) sollten Sie benutzen wenn Sie nur über ein Laufwerk verfügen und Dateien von einer Diskette zu einer anderen übertragen wollen. Bedenken Sie, daß Ihr Laufwerk als Laufwerk I gekennzeichnet sein muß. Bei einem Laufwerk müssen Sie die Originaldiskette und die Zieldiskette per Hand wechseln, um einen Datentransfer zu gewährleisten. Bei längeren Dateien kann es also passieren, daß Sie die Disketten öfters wechseln müssen.

Beispiel I:

```
SELECT ITEM OR [RETURN] FOR MENU
O [RETURN]
NAME OF FILE TO MOVE?
DOSEX.BAS [RETURN]
INSERT SOURCE DISK, TYPE RETURN
[RETURN]
INSERT DESTINATION DISK, TYPE RETURN
[RETURN]
```

Beispiel 2:

```
SELECT ITEM OR [RETURN] FOR MENU
O [RETURN]
NAME OF FILE TO MOVE?
TEST? [RETURN]
INSERT SOURCE DISK, TYPE RETURN
[RETURN]

  COPYING---D1:TEST1
INSERT DESTINATION DISK, TYPE RETURN
[RETURN]

INSERT SOURCE DISK, TYPE RETURN
[RETURN]

  COPYING---D1:TEST2
INSERT DESTINATION DISK, TYPE RETURN
[RETURN]

INSERT SOURCE DISK, TYPE RETURN
[RETURN]
```

In Beispiel 3 wurden sowohl die Dateinamen, als auch die Extender durch "Wild Cards" ersetzt. Damit werden alle Dateien, mit Ausnahme der System-Dateien (Extender .SYS), vom DOS kopiert. Es dürften demnach nur die drei Dateien MEM.SAV, TEST1 und TEST2 kopiert worden sein.

Anmerkung: Es geht schneller, eine neue Datei MEM.SAV auf der Diskette anzulegen, als die Datei zu kopieren.

Beispiel 3:

```
SELECT ITEM OR [RETURN] FOR MENU
O [RETURN]
NAME OF FILE TO MOVE?
*.* [RETURN]
INSERT SOURCE DISK, TYPE RETURN
[RETURN]

  COPYING-D1:MEM.SAV
INSERT DESTINATION DISK, TYPE RETURN
[RETURN]

INSERT SOURCE DISK, TYPE RETURN
[RETURN]
```

```
  COPYING---D1:TEST1
INSERT DESTINATION DISK, TYPE RETURN
[RETURN]

INSERT SOURCE DISK, TYPE RETURN
[RETURN]

  COPYING---D1:TEST2
INSERT DESTINATION DISK, TYPE RETURN
[RETURN]

INSERT SOURCE DISK, TYPE RETURN
[RETURN]
```

P. Format Single

Verwenden Sie diese Option, um eine Diskette in single density (einfache Dichte) zu formatieren. Wenn Sie nur über ein ATARI XF551 Diskettenlaufwerk verfügen, müssen Sie anstelle von Option I im DOS-Menü diese Funktion anwählen, um eine Diskette in single density zu formatieren.

Denken Sie daran, daß durch Formatieren einer Diskette sämtliche Daten gelöscht werden, die zu einem früheren Zeitpunkt bereits auf dieser Diskette gespeichert wurden.

Das Verfahren bei FORMAT SINGLE ist genau dasselbe wie bei FORMAT DISK:

```
SELECT ITEM OR [RETURN] FOR MENU
P [RETURN]
WHICH DRIVE TO FORMAT?
1 [RETURN]
TYPE "Y" TO FORMAT DISK 1
Y [RETURN]
```

Kapitel 4: Die Verwendung von BASIC-Befehlen mit DOS 2.5

BASIC-Befehle, die mit DOS verwendet werden können

Bevor Sie Informationen über die BASIC-Befehle erhalten, die Sie mit DOS 2.5 verwenden können, sollten Sie wissen, welche Auswirkungen diese Befehle auf Programme haben, die Sie speichern und laden können. Im nachfolgenden Absatz werden die beiden Datei-Arten beschrieben, die BASIC-Programme enthalten können.

Komprimierte und nicht komprimierte Dateien

Die erste Datei-Art, die im nicht komprimierten ("untokenized") Format abgelegt wird, enthält standardisierte ATASCII Text-Zeichen und kann wie ein ausgedrucktes BASIC-Programm gelesen werden. Bei diesen Programmen wird die dazugehörige Variablen-Tabelle nicht bei jedem Speicher- und Ladevorgang gehalten. Über die Variablen-Tabelle wird der Variablenname einer Speicherstelle zugeordnet, in der die Werte für diese Variable abgelegt sind. Zum Speichern und Laden einer Datei im nicht komprimierten Format werden die Befehle LIST und ENTER verwendet.

Eine komprimierte ("tokenized") Datei ist eine verdichtete Version eines BASIC-Programmes. Sie enthält anstelle von ATASCII-Zeichen 1-Byte Tokens zur Darstellung von BASIC-Befehlen. Komprimierte Dateien können mit den Befehlen LOAD und SAVE von Diskette geladen oder darauf gespeichert werden. Komprimierte Dateien sind kürzer und können schneller geladen werden als ATASCII-Dateien. Aus diesem Grund speichern Programmierer ihre Dateien in der Regel im komprimierten Format.

Normalerweise wird die Variablen-Tabelle während einer Programmentwicklung durch nicht verwendete Variablennamen unübersichtlich. Wenn Sie in einer Programmzeile eine Variable verwenden und danach diese Variable umbenennen oder die Programmzeile wieder löschen, bleibt der Variablenname dennoch in der Variablen-Tabelle enthalten. Um nicht mehr benötigte Variablennamen aus der Tabelle zu entfernen, verfahren Sie wie folgt:

1. Laden Sie Ihr Programm mit LOAD (siehe "LOAD" im nächsten Absatz).

2. Listen Sie das Programm mit LIST auf (siehe "LIST" im nächsten Absatz).

3. Geben Sie NEW (RETURN) ein.

4. Laden Sie Ihr Programm nun mit ENTER von Diskette ein (siehe "ENTER"). Die Variablen-Tabelle enthält nun nur noch die Variablennamen, die Sie in Ihrem Programm auch ansprechen.

LOAD (LO.)

Format: LOAD Dateiangabe
Beispiel: LOAD "D1:DOSEX.BAS" (RETURN)

Mit diesem Befehl wird eine Datei von einer in das Laufwerk eingelegten Diskette in den für den Benutzer freigegebenen Bereich des RAM geladen. Bevor Sie mit diesem Befehl eine Datei DOSEX.BAS laden können, muß diese Datei unter Verwendung des BASIC-Befehles SAVE bereits einmal gespeichert worden sein. Mit LOAD können nur komprimierte Dateien geladen werden.

Der Befehl LOAD kann auch zum Verketten von Programmen verwendet werden. Wenn eines Ihrer Programme zu lang ist, um in dem für den Benutzer freigegebenen Bereich des RAM abzulaufen, kann es mit LOAD in zwei Dateien aufgeteilt werden. Geben Sie einfach in der letzten Programmzeile der ersten Programmdatei LOAD ein, wie im nachstehenden Beispiel beschrieben wird. Sobald die Programmausführung auf den Befehl LOAD trifft, wird automatisch der nächste Teil des Programmes von Diskette nachgeladen. Dabei muß das zweite Programm jedoch alleine ausführbar sein und darf keine Verbindung zu Variablen oder Daten in dem im RAM liegenden ersten Programmteil aufweisen. Der zweite Programmteil wird erst ausgeführt, wenn Sie den Befehl RUN eingeben und (RETURN) drücken. Damit wird der erste Programmteil mit allen Variablen aus dem Speicher entfernt. (Ein weiteres Beispiel hierzu finden Sie unter "RUN".)

```
100 REM PROGRAMM VERKETTEN
110 LOAD "D1:CHAIN.BAS"
```

SAVE (S.)

Format: SAVE Dateiangabe
Beispiel: SAVE "D1:BEISP2.BAS" (RETURN)

Mit diesem Befehl wird ein Programm unter dem angegebenen Dateinamen auf Diskette gespeichert.

SAVE ist das Gegenstück zum Befehl LOAD. Programme werden mit SAVE im komprimierten Format abgelegt.

LIST (L.)

Formate: LIST Dateiangabe, Zeilennummer (,Zeilennummer) Gerät
Beispiele: LIST "D:DATEI.LST"
LIST "P:"
LIST "P:",10,100

Eine Einsatzmöglichkeit von LIST hat große Ähnlichkeit mit dem Befehl SAVE - Sie können damit ein Programm aus dem Benutzerbereich des RAM herausnehmen und unter einem beliebigen, von Ihnen zugewiesenen Namen auf Diskette ablegen (wie im obenstehenden Beispiel gezeigt).Allerdings wird hier das Programm im Standard ATASCII-Format, und nicht in komprimierter Form gespeichert. Die Unterschiede in der Formatierung zu speichernder Daten ergeben eine größere Flexibilität bei der Verwendung von LIST im Vergleich zu SAVE. Wie Sie aus den nachstehenden Beispielen ersehen können, können mit LIST sowohl einzelne Geräte (z.B. P:, E:, C:, D:, D2: usw.), als auch Zeilennummern für die Ausgabe auf einem bestimmten Gerät (z.B. "P:",100, 200) angegeben werden.

ENTER (E.)

Format: ENTER Dateiangabe
Beispiel: ENTER "D:LIST2.LST"

Mit Hilfe dieses Befehles wird eine angegebene Datei von Diskette in den Benutzerbereich des RAM eingebracht. Dabei wird das Programm in nicht komprimierter Form geladen und beim Empfang der Daten interpretiert. Mit ENTER wird - im Gegensatz zu LOAD - das im RAM befindliche BASIC-Programm nicht gelöscht, sondern mit dem neu geladenen Programm verknüpft. Existieren in den beiden Programmteilen gleiche Zeilennummern, werden die Zeilennummern des im RAM befindlichen Programmes durch die Zeilennummern des mit ENTER nachgeladenen Programmes ersetzt.

RUN

Format: RUN Dateiangabe
Beispiel: RUN "D2:DATEI.BAS"

Mit diesem Befehl wird die angegebene Programm-Datei in den Speicher geladen und ausgeführt. Damit handelt es sich um eine Befehlskombination aus LOAD und RUN, wobei RUN nur für die Ausführung komprimierter Dateien verwendet werden kann. Deshalb können Sie keinen Befehl wie beispielsweise RUN "D2:LIST.LST" ausführen lassen.

Um Programme zu verketten und dann den zweiten Programmteil automatisch nachzuladen und auszuführen, kann der Befehl RUN "D:Dateiangabe" in die letzte Zeile des ersten Programmsegmentes geschrieben werden. Allerdings muß auch hier das zweite Programmsegment allein lauffähig sein und darf nicht von Daten oder Variablen aus dem ersten Programmsegment abhängen, da das erste Segment beim Laden des zweiten Programmteiles aus dem RAM gelöscht wird. Vergewissern Sie sich also vor der Ausführung derart verketteter Programme, daß das erste Segment zuvor auf einer Diskette abgespeichert wurde.

Steuerblöcke für Ein- und Ausgabe

Eine I/O (Eingabe/Ausgabe) Operation wird durch einen I/O - Steuerblock (IOCB) kontrolliert. Ein IOCB ist die Spezifikation der I/O Operation und besteht aus dem I/O-Typ, der Pufferlänge, der Pufferadresse und zwei weiteren Hilfsvariablen, von denen die zweite zumeist 0 ist. ATARI BASIC stellt acht IOCBs zur Verfügung und definiert drei davon wie folgt:

IOCB #0 wird von BASIC für I/O an E: verwendet
IOCB #6 wird von BASIC für I/O an S: verwendet
IOCB #7 wird von BASIC für LPRINT-, CLOAD- und SAVE-Befehle verwendet.

IOCB #1 bis #5 können beliebig verwendet werden. Die vordefinierten IOCBs dagegen sollten nicht angesprochen werden, sofern nicht ein Programm spezielle Verwendung für eine der beschriebenen Definitionen hat. IOCB #0 kann niemals über ein BASIC-Programm geöffnet oder geschlossen werden.

Jedem I/O Befehl muß ein entsprechender IOCB zugeordnet werden. I/O Befehle, die zusammen mit DOS 2.5 verwendet werden können, sind:

OPEN / CLOSE
INPUT / PRINT
PUT / GET
STATUS
XIO

Die Befehle "OPEN" und "CLOSE"

OPEN (O.)

Format: OPEN #IOCB,AEXP1,AEXP2,Dateiangabe
Beispiel: 100 OPEN #2,8,0,"D1:ATARI800.BAS"

Über OPEN wird veranlaßt, daß ein spezifizierter IOCB mit der dazugehörigen Geräte-Steuerung verbunden wird, sämtliche CIO-abhängigen Kontroll-Variablen initialisiert (siehe Glossar), sowie alle Geräte-spezifischen Optionen an den Geräte-Treiber übergeben werden. Die Parameter in dieser Anweisung werden folgendermaßen definiert:

Zwingend vorgegebenes Zeichen, das vom Anwender eingegeben werden muß

IOCB Eine Zahl zwischen 1 und 7, die sich auf ein Gerät bzw. eine Datei beziehen muß

AEXP1 Eine Zahl, über die die Art der auszuführenden Operation festgelegt wird:

Code 4 = Eingabe-Operation - positioniert den Dateizeiger am Anfang der Datei

Code 6 = Eingabe-Operation für Diskettenverzeichnis Kompatibel mit DOS 2.0S

Code 7 = Eingabe-Operation für Diskettenverzeichnis, mit DOS 2.5 Informationen

Code 8 = Ausgabe-Operation - positioniert den Dateizeiger am Anfang der Datei

Code 9 = Dateiende - Append-Operation - positioniert den Dateizeiger am Ende der Datei. Mit Code 9 kann eine Programm-Eingabe über den Bildschirm-Editor vorgenommen werden, die der Anwender nicht mit RETURN bestätigen muß

Code 12= Ein- und Ausgabe-Operation - positioniert den Dateizeiger am Anfang der Datei

AEXP2 Geräte-abhängiger Hilfscode

Dateiangabe Spezifische Bezeichnung der Datei (siehe auch Abschnitt 3, Definition von Dateiangaben).

In dem Beispiel OPEN #2,8,0,"D1:ATARI800.BAS" wird IOCB #2 für eine Ausgabe an eine Datei mit dem Namen ATARI800 auf der in Laufwerk 1 befindlichen Diskette geöffnet. Existiert noch keine Datei mit diesem Namen auf der eingelegten Diskette, legt DOS eine neue Datei an. Befindet sich auf der Diskette bereits eine Datei mit dem angegebenen Namen, wird diese durch den Befehl OPEN gelöscht und stattdessen eine neue Datei mit diesem Namen angelegt. Wurde der bezeichnete IOCB bereits geöffnet, erhalten Sie die Fehlermeldung ERROR-129 (Datei bereits geöffnet) auf dem Bildschirm.

CLOSE (CL.)

Format: CLOSE #IOCB
Beispiel: 300 CLOSE #2

Der Befehl CLOSE schließt den IOCB, der zuvor über OPEN für Lese- oder Schreib-Operationen geöffnet wurde. Die Zahl hinter dem obligatorischen Nummernzeichen (#) muß der IOCB-Referenzzahl entsprechen, die in der OPEN-Anweisung verwendet wurde (siehe nachstehendes Beispiel). Ein IOCB kann niemals für mehr als ein Gerät gleichzeitig angegeben werden. Wenn Sie eine Datei schließen wollen, die bereits zu einem früheren Zeitpunkt geschlossen wurde, erhalten Sie keine Fehlermeldung auf dem Bildschirm.

```
10 OPEN #1,8,0,"D:DATEI.BAS"
20 CLOSE #1
```

Anmerkung: Mit dem Befehl END werden alle derzeit geöffneten Dateien geschlossen (mit Ausnahme von IOCB #0).

Die Befehle "INPUT" und "PRINT"

INPUT (I.)

Format: INPUT [#IOCB ,] Arithm.Var.,Arithm.Var ;] String Var. , String Var.
Beispiele: 100 INPUT #2,X,Y
100 INPUT #2,N$

Dieser Befehl wird verwendet, um Daten (in numerischer Form oder als String) von einem angegebenen Gerät abzurufen. INPUT ist das Gegenstück zu PRINT. Wenn Sie diesen Befehl ohne Angabe einer IOCB-Nummer verwenden, wird angenommen, daß die Daten vom vorbestimmten Gerät (E:) abgerufen werden sollen. INPUT arbeitet mit Record-I/O (siehe auch unter PRINT).

In dem nachfolgenden Beispielprogramm für INPUT/PRINT erhält der Anwender in Zeile 35 die Möglichkeit, Daten über die Tastatur (vorbestimmte Geräte-Angabe) einzugeben. In Zeile 70 wird mit der INPUT-Anweisung der Inhalt der eingegebenen Zeichenkette von der geöffneten Datei eingelesen.

```
 5 REM **ANLEGEN EINER DATEN-DATEI**
 7 REM **OPEN MIT 8 ERSTELLT EINE DATEN-DATEI**
10 OPEN #1,8,0,"D:SCHREIB.DAT"
20 DIM WRT$(60)
30 ? "SATZ MIT MAXIMAL 60 BUCHSTABEN EINGEBEN"
35 INPUT WRT$
38 REM **SCHREIBT DIE DATEN AUF DISKETTE**
40 PRINT #1:WRT$
45 REM **SCHLIESST DATEN-DATEI**
50 CLOSE #1
55 REM **DATEN-DATEI OEFFNEN ZUM LESEN**
58 REM ** OPEN MIT 4 GILT FUER READ-ONLY DATEIEN** 60 OPEN #1,4,0,"D-
:SCHREIB.DAT"
65 REM **LIEST DATEN VON DISKETTE**
70 INPUT #1,WRT$
75 REM **DRUCKT DATEN**
80 PRINT WRT$
85 REM **SCHLIESST DATEN-DATEI**
90 CLOSE #1
```

PRINT (PR. oder ?)

Format: PRINT #IOCB ; [Ausdruck]...[Ausdruck]
Beispiele: 100 PRINT #2,X,Y
100 PRINT #2,A$
100 ? O$
100 PRINT "X = ",X

Mit diesem Befehl wird ein (arithmetischer oder String) Ausdruck an das geöffnete Gerät mit der gleichen IOCB-Nummer gegeben.

Ist keine IOCB-Nummer angegeben, schreibt das Betriebssystem den Ausdruck auf den Bildschirm (vordefiniertes Gerät). Wird die Information an ein Gerät geleitet, das derzeit nicht geöffnet ist, erhalten Sie die Fehlermeldung ERROR-133 auf dem Bildschirm.

PRINT entwickelt sogenannte Record-I/Os. Records, auch Datensätze genannt, sind Zusammenfassungen von Bytes, die durch Zeilenende-Zeichen (9B Hex) voneinander abgetrennt sind. Records können beliebig lang sein. Die Länge eines Datensatzes kann durch die Länge einer Zeichenkette, die an eine Disketten-Datei ausgegeben werden soll, oder durch das Format einer arithmetischen Variablen festgelegt werden. Ein Datensatz kann auch die Länge eines Strings sein, der über Tastatur eingegeben und mit einem RETURN abgeschlossen wurde.

In der Regel kann die INPUT-Anweisung keine Datensätze lesen, die länger als 255 Zeichen sind. Wenn Sie einen Datensatz mit PRINT auf Diskette schreiben und diesen später über INPUT wieder einlesen lassen wollen, sollten Sie sinnvollerweise die Länge der mit PRINT abgelegten Records auf maximal 255 Zeichen begrenzen.

Direktzugriff auf Dateien mit den Befehlen NOTE/POINT

NOTE (NO.)

Format: NOTE #IOCB,Arith.Var.,Arith.Var.
Beispiel: NOTE #2,A,B

Dateien werden normalerweise sequentiell erstellt, wobei der Zugriff auf die Daten der Reihe nach - vom Anfang zum Ende der Datei - erfolgt. Wollen Sie auf Datensätze in einer Datei in nicht sequentieller Folge zugreifen, müßten Sie die Datei in der normalen Folge durchsuchen und an dem gesuchten Datensatz anhalten, sofern Sie keine Möglichkeit einer direkten Adressierung des gesuchten Datensatzes haben.

Da das sequentielle Durchsuchen einer Datei sehr zeitaufwendig ist, bietet DOS 2.5 die beiden Befehle NOTE und POINT an, mit deren Hilfe Sie beliebig auf Dateien Zugriff nehmen können. Um einen Datensatz aufzufinden, ohne sämtliche voranstehenden Records durchsuchen zu müssen, müssen Sie dem Computer genau angeben, auf welchen Datensatz Sie zugreifen wollen. Hierzu ist erforderlich, den betreffenden Datei-Sektor zu markieren, indem Sie vor jedem Schreibvorgang einen NOTE-Befehl geben und den daraufhin übermittelten Wert in einer Tabelle ablegen lassen.

Mit NOTE wird der Wert des aktuellen Dateizeigers für die Datei mit der vorgegebenen IOCB-Nummer ermittelt. Über den Dateizeiger wird genau angegeben, welches Byte als nächstes gelesen bzw. geschrieben werden soll. Dieser Befehl legt die absolute Sektor-Kennziffer der Diskette in der ersten arithmetischen Variable, und die aktuelle Byte-Zahl in der zweiten Variable ab. Sektor-Kennziffern liegen bei Disketten mit einfacher Dichte im Bereich zwischen 1 und 719, bei Disketten mit erweiterter Dichte zwischen 1 und 1023. Als Byte-Zahlen werden Werte von 0 bis 124 zugelassen.

Aus dem nächsten Beispielprogramm können Sie ersehen, wie unter Verwendung von NOTE Eingaben über Tastatur an einer angegebenen Stelle in einer Datei abgelegt werden.

Nachfolgend sehen Sie ein Beispielprogramm, in dem NOTE dazu verwendet wird, Tastatur-Eingaben in einer Random-Access Disketten-Datei abzulegen. Schreiben Sie das Programm-Listing ab und geben Sie ein paar Zahlen ein. Beenden Sie das Programm durch Drücken von (RETURN). In dem Beispiel werden Zahlenwerte eingegeben - Sie könnten jedoch auch Zeichenketten mit maximal 40 Zeichen als A$ eingeben.

```
 1 REM DEMO FUER DIE ANWEISUNG NOTE
 2 REM DIESES PROGRAMM LIEST ZEILEN MIT DATEN
 3 REM VON TASTATUR EIN UND SPEICHERT
 4 REM SIE UNTER D:DATEI.DAT AUF DISKETTE
 5 REM ZEIGER WERDEN UNTER D:ZEIGER.DAT GESPEICHERT
20 DIM A$(40)
25 OPEN #1,8,0,"D:DATEI.DAT"
27 OPEN #2,8,0,"D:ZEIGER.DAT"
30 REM LIEST ZEILEN MIT DATEN VON TASTATUREINGABEN
40 INPUT A$
41 PRINT A$
42 REM RETURN OHNE EINGABE = STOP
45 IF LEN(A$)=0 THEN 100
50 NOTE #1,X,Y
55 REM SPEICHERT ZEILE MIT DATEN
60 PRINT #1,A$
61 REM SPEICHERT ZEIGER AM ANFANG DER DATEN-ZEILE
65 PRINT #2,X,",",Y
70 PRINT "SEKTOR #= ",X,"BYTE #= ",Y
90 GOTO 40
95 REM ANZEIGE DATEIENDE
100 PRINT #2,0,",",0
110 END
```

Die Ausführung des Beispielprogrammes erfolgte auf einer Diskette, auf der sich die Dateien DOS.SYS, DUP.SYS und MEM.SAV befanden. In Ihrem Programmlauf könnten die Angaben der Sektoren- und Byte-Zahlen unterschiedlich sein. Die Zahleneingaben im Beispiel lauteten 45, 55, 75, 80, 90, 100 und 110.

```
45
SEKTOR # = 145          BYTE # = 9 55
SEKTOR # = 145          BYTE # = 12
75
SEKTOR # = 145          BYTE # = 15
80
SEKTOR # = 145          BYTE # = 18
90
SEKTOR # = 145          BYTE # = 21
100
SEKTOR # = 145          BYTE # = 24
110
SEKTOR # = 145          BYTE # = 28
```

POINT (P.)

Format: POINT #IOCB, Arith.Var.,Arith.Var.
Beispiel: 100 POINT #2,A,B

POINT ist das Gegenstück zu NOTE. Mit diesem Befehl wird der Dateizeiger auf einen beliebigen Wert gesetzt, der über die arithmetischen Variablen bestimmt wird. POINT wird benutzt, wenn spezifizierte Datei-Positionen (Sektor und Byte) ins RAM gelesen werden sollen. Die erste arithmetische Variable spezifiziert (bzw. zeigt auf) die Sektor-Kennziffer, die zweite arithmetische Variable gibt die Byte-Zahl an, in die das nächste Byte gelesen bzw. geschrieben werden soll. Wie auch bei dem Befehl NOTE gilt für die Sektor-Kennziffer ein Bereich von 1 bis 719 bei single density Disketten und 1 bis 1023 bei enhanced density Disketten. Der Bereich für die Byte-Zahlen liegt zwischen 0 und 124. Wenn Sie den Zeiger auf einen Bereich außerhalb einer geöffneten Datei richten, erhalten Sie eine Fehlermeldung "Dateinummer nicht übereinstimmend". Im nachfolgenden Programmbeispiel sehen Sie, wie mit dem Befehl POINT Daten gelesen werden können, die von dem zuvor erstellten Beispielprogramm für NOTE generiert wurden.

Bei der Programmausführung werden die Tastatureingaben für Sektoren und Bytes in entgegengesetzter Reihenfolge ausgegeben, in der sie auf Diskette geschrieben wurden.

Nachdem Sie die Beispielprogramme für NOTE und POINT eingegeben haben, starten Sie zuerst das Programm NOTE, und danach das Programm POINT, ohne zwischenzeitlich einen Diskettenwechsel vorzunehmen.

```
  1 REM DEMO FUER DIE ANWEISUNG POINT
  2 REM DIESES PROGRAMM LIEST DIE MIT DEM NOTE-PROGRAMM
  3 REM ERSTELLTE DATEI EIN UND GIBT DIE ZEILEN IN
  4 REM UMGEKEHRTER REIHENFOLGE AUS 10 DIM B(20,1)
20 DIM A$(40)
25 REM OEFFNEN DER DATEN-DATEI
30 OPEN #1,4,0,"D:DATEI.DAT"
35 REM OEFFNEN DER ZEIGER-DATEI
40 OPEN #2,4,0,"D:ZEIGER.DAT"
45 REM LIEST ZEIGER IN EIN ARRAY EIN
50 FOR I=0 TO 20
60 INPUT #2,X,Y
70 B(I,0)=X:B(I,1)=Y
80 IF X=0 AND Y=0 THEN LAST=I:I=20
90 NEXT I
95 REM GIBT DATEI IN UMGEKEHRTER FOLGE AUS
100 FOR I=LAST-1 TO 0 STEP -1
110 X=B(I,0):Y=B(I,1)
120 POINT #1,X,Y,
130 PRINT "SEKTOR #= ",X,"BYTE #= ",Y
140 INPUT #1,A$
150 PRINT A$
160 NEXT I
```

Die Programmausführung verläuft wie folgt:

```
SEKTOR # = 145          BYTE # = 28 110
SEKTOR # = 145          BYTE # = 24
100
SEKTOR # = 145          BYTE # = 21
90
SEKTOR # = 145          BYTE # = 18
80
SEKTOR # = 145          BYTE # = 15
75 SEKTOR # = 145       BYTE # = 12
55 SEKTOR # = 145       BYTE # = 9
45
```

Die Befehle "PUT" und "GET"

PUT (PU.)

Format: PUT #IOCB,Arith.Ausdruck
Beispiel: 100 PUT #6,ASC("A")

Der Befehl PUT dient dazu, ein einzelnes Byte (= ein Wert zwischen 0 und 155) an das Gerät weiterzugeben, das durch die IOCB-Nummer spezifiziert wurde. In dem nachfolgenden Beispielprogramm wird PUT dazu verwendet, um jeden Wert, den Sie eingeben, in einem auf A(50) dimensionierten Array abzulegen. Sie können maximal 50 Zahlen eingeben, wobei die Werte zwischen 0 und 255 liegen müssen. Der Befehl PUT wird benutzt, um Daten-Dateien zu erstellen oder um Daten in bereits existierende Dateien einzufügen.

GET (GE.)

Format: GET #IOCB,Arith.Var.
Beispiel: 100 GET #2,X

Mit dem Befehl GET wird ein einzelnes Byte von dem über die IOCB-Nummer angegebenen Gerät in eine von Ihnen spezifizierte Variable eingelesen. GET ermöglicht das Auffinden von Bytes, die über PUT gespeichert wurden.

Das nachstehende Beispielprogramm demonstriert die Einsatzmöglichkeiten von PUT und GET. In den Zeilen 130 bis 230 wurde ein Null-Byte (das von Ihnen eingegeben wird) dazu verwendet, um den Abschluß der Dateneingabe anzuzeigen. (Normalerweise würde eine TRAP-Anweisung gesetzt, um das Dateiende zu bestimmen.)

```
 10 REM **PUT/GET BEISPIEL**
 20 DIM A(50),A$(10)
 30 ? "BEISPIEL ZUM ABLEGEN UND EINLESEN AUF DISKETTE"
 40 ? "GEBEN SIE 'SCHREIBEN' ODER 'LESEN' EIN !":INPUT A$:?
 50 IF A$="LESEN" THEN 170
 60 IF A$<>"SCHREIBEN" THEN PRINT "?":GOTO 40
 70 REM ROUTINE ZUM SCHREIBEN DER DATEN
 80 OPEN #1,8,0,"D1:BEISP1.DAT"
 90 ? "GEBEN SIE EINE ZAHL KLEINER 256 EIN":PRINT "0 ZUM BEENDEN"
    : INPUT X
 95 REM SCHREIBT ZAHL IN DIE DATEI
100 PUT #1,X
110 IF X=0 THEN CLOSE #1:GOTO 130
120 GOTO 90
130 GRAPHICS 0: ?: ? "DATEN IN DATEI EINLESEN?":INPUT A$: ?
140 IF A$="NEIN" THEN END
150 IF A$<> "JA" THEN 130
160 REM ROUTINE ZUM AUSLESEN DER DATEN
170 OPEN #2,4,0,"BEISP1.DAT"
180 FOR E=1 TO 50
185 REM LIEST ZAHL(EN) AUS DER DATEI
190 GET #2,G:A(E)=G
200 IF G=0 THEN GOTO 230
210 PRINT "BYTE #",E,"=",G
220 NEXT E
230 CLOSE #2
```

Beachten Sie, daß INPUT/PRINT und PUT/GET zwei unterschiedliche I/O-Befehle sind, die untereinander nicht kompatibel sind. PRINT fügt EOL (Zeilenende) - Zeichen zwischen die einzelnen Records ein und INPUT benutzt diese Zeichen, um einen Datensatz zu bestimmen. GET und PUT dagegen lesen und schreiben Bytes aus bzw. in Dateien, ohne sie durch EOL-Zeichen voneinander abzugrenzen. Eine Datei, die unter Verwendung von PUT-Anweisungen erstellt wurde, sieht aus wie ein einziger, langer Datensatz, solange Sie nicht selbst ein EOL (9B Hex) - Zeichen in die Datei einfügen.

Nachdem Sie das PUT/GET Beispielprogramm eingegeben haben, geben Sie RUN ein und drücken (RETURN). Geben Sie die Werte 2, 5, 67, 54 und 68 ein. Sobald Sie die Programmausführung starten, werden die eingegebenen Zahlen zusammen mit der Angabe, an welcher Position das jeweilige Byte abgelegt ist, ausgegeben. Beispiel:

BYTE #1 = 2
BYTE #2 = 5
BYTE #3 = 67
BYTE #5 = 68

Der Befehl "STATUS"

STATUS (ST.)

Format: STATUS #IOCB,Arith.Var.
Beispiel: 100 STATUS #5,ERROR

Der Befehl STATUS wird verwendet, um die Bedingung (den Status) einer Datei festzulegen. Es handelt sich hierbei um einen CIO-Befehl, über den einige Ursachen für das Auftreten eines Fehlers geprüft werden. Die erste Gruppe möglicher Fehlerursachen wird wie folgt überprüft:

Sektor-Puffer verfügbar?	Falls nicht	- ERROR-161
Gültige Geräte-Nummer?	Falls nicht	- ERROR-20
Gültiger Dateiname?	Falls nicht	- ERROR-170
Datei auf Diskette existent?	Falls nicht	- ERROR-170
Datei gesperrt?	Falls ja	- ERROR-167

Sie können auch alle I/O-Fehler für den seriellen Bus über eine STATUS-Anweisung identifizieren lassen. Dabei handelt es sich um folgende Fehlergruppen:

Antwortfrist abgelaufen	ERROR-138
Geräte-Angabe ungültig	ERROR-139
Serieller Begrenzungsfehler	ERROR-140
Überlauf Datenbegrenzung im seriellen Bus	ERROR-141
Prüfsummen-Fehler im seriellen Bus	ERROR-142
Unpassende Geräteausgabe	ERROR-144

Bevor Sie einen STATUS-Befehl geben können, müssen Sie die entsprechende Datei geöffnet haben. Sinnvollerweise sollten Sie die XIO-Befehlsform für diesen Befehl verwenden, da dieses Format zum Einen zuverlässiger ist, und zum Anderen ein aufgefundener Fehler mit einem spezifischen Dateinamen versehen werden kann.

Der Einsatz des XIO-Befehles in DOS Menüoptionen

XIO (X.)

Format: XIO Befehlnr, #IOCB, arith.Ausdruck1,2,Dateiangabe
Beispiel: 100 XIO 3,#6,4,0,"D:TEST.BAS"

Der XIO-Befehl ist eine allgemeine INPUT/OUTPUT-Anweisung für spezielle Operationen. Er wird verwendet, wenn Sie eine Funktion entwickeln wollen, die normalerweise über die DOS Menüoptionen erarbeitet wird.

Die einzelnen XIO-Befehle dienen zum Öffnen und Schließen von Dateien, Lesen und Schreiben von Datensätzen oder Zeichen, Bestimmen von Positionen in einer Datei für Lese- und Schreibvorgänge, zum Umbenennen, Löschen, Sperren und Entsperren von Dateien. Beachten Sie dabei, daß ein XIO-Aufruf eine Dateiangabe erfordert.

Die Befehl-Kennziffer (Befehlnr.) wird zur Spezifikation der zu entwickelnden Operation verwendet.

Befehlnr.	Operation	Beispiel
13	STATUS-Abfrage	XIO 13,#1,0,0,"D:TEST.BAS"
32	Umbenennen (RENAME)	XIO 32,#1,0,0,"D:ALT,NEU"
33	Löschen (DELETE)	XIO 33,#1,0,0,"D:ALT.BAS"
35	Datei sperren (LOCK)	XIO 35,#1,0,0,"D:ATARI.BAS"
36	Datei entsperren (UNLOCK)	XIO 36,#1,0,0,"D:DOSEX.BAS"
253	Formatieren (FORMAT DISK)	XIO 253,#1,0,0,"D1:" Formatierung single density
254	Formatieren (FORMAT DISK)	XIO 254,#1,0,0,"D1:" Formatierung enhanced density

Anmerkung: Geben Sie beim Umbenennen von Dateien den Geräte-Namen nicht zweimal an (d.h. schreiben Sie nicht "D:ALT,D:NEU")

Nachfolgend sehen Sie ein kleines DOS-Programm, mit dem Sie innerhalb von BASIC Diskettendateien manipulieren können. Normalerweise müßten Sie zum Umbenennen, Löschen, Sperren oder Entsperren von Dateien BASIC durch Eingeben von "DOS" verlassen, um auf die Betriebssystem-Ebene zurückzukommen. Durch Aufrufen des DOS-Menüs wird allerdings der gesamte Inhalt des RAM gelöscht, wodurch Sie ein gerade in Ausarbeitung befindliches BASIC-Programm zerstören würden. Unter Verwendung des XIO-Befehles können Sie derartige Funktionen entwickeln, ohne auf die DOS-Ebene zurückkehren zu müssen.

```
 10 REM BEISPIELPROGRAMM FUER ZUGRIFF AUF DOS-FUNKTIONEN
 20 REM INNERHALB VON BASIC
 30 DIM FILE$(40),NAME2$(20)
 40 REM DISKETTENVERZEICHNIS LESEN
 50 GRAPHICS 0
 60 TRAP 100:OPEN #1,6,0,"D:*.*"
 70 FOR I=1 TO 64
 80 INPUT #1,FILE$
 90 ? FILE$:NEXT I
100 CLOSE #1
110 ? "WOLLEN SIE EINE DATEI (1) UMBENENNEN"
120 ?: ? "(2) LÖSCHEN":? :? "(3) SPERREN"
130 ? :?"(4) ENTSPERREN":?:?"(5) VERLASSEN"
140 ?:? "GEBEN SIE EINE ZAHL (1-5) EIN",:INPUT N
150 IF N<1 OR N>5 THEN 140
160 ON N GOSUB 200,300,400,500,170:GOTO 50
170 END
198 REM UNTERROUTINE ZUM UMBENENNEN EINER DATEI
199 REM MIT XIO 32
200 ? "DATEINAMEN FUER NEU ZU BENENNENDE DATEI WIE FOLGT
    EINGEBEN: D:DATEINAME.EXT"
210 INPUT FILE$
220 ? "NEUEN DATEINAMEN FUER "FILE$," IN DER FORM DATEINAME.EXT
    EINGEBEN"
230 INPUT NAME2$
235 FILE$(LEN(FILE$)+1)=","
237 FILE$(LEN(FILE$)+1)=NAME2$
240 XIO 32,#1,0,0,FILE$
250 RETURN
298 REM UNTERROUTINE ZUM LOESCHEN EINER DATEI
299 REM MIT XIO 33
300 ?: ? "ZU LOESCHENDE DATEI WIE FOLGT EINGEBEN:
    D:DATEINAME.EXT"
310 INPUT FILE$
320 ?: ? "WOLLEN SIE DIE DATEI",FILE$, "WIRKLICH LOESCHEN?
    (J/N)",
330 INPUT NAME2$
340 IF NAME2$(1,1)<>"J" THEN 360
```

```
350 XIO 33,#1,0,0,FILE$
360 RETURN
398 REM SPERREN EINER DATEI MIT XIO 35
400 ? "DATEINAMEN DER ZU SPERRENDEN DATEI WIE FOLGT
    EINGEBEN: D:DATEINAME.EXT"
410 INPUT FILE$
420 XIO 35,#1,0,0,FILE$
430 RETURN
498 REM ENTSPERREN EINER DATEI MIT XIO 36
500 ? "DATEINAMEN ZUM ENTSPERREN FOLGENDERMASSEN EINGEBEN:
    D:DATEINAME.EXT"
510 INPUT FILE$
520 XIO 36,#1,0,0,FILE$
530 RETURN
```

Zugriff auf zerstörte Dateien

Dateien können auf zweierlei Arten zerstört werden. Im einen Fall wurde lediglich das Diskettenverzeichnis unlesbar gemacht, wodurch auf alle Angaben über die Datei (Dateiname, Verzeichniszeiger, Anzahl der Sektoren in der Datei) kein Zugriff mehr genommen werden kann. Die zweite Möglichkeit ist, daß die Datei selbst zerstört ist. (Nähere Informationen über den Zugriff auf zerstörte Dateien finden Sie in dem Abschnitt DISKFIX.COM in Anhang L.)

Wurde das Diskettenverzeichnis unlesbar gemacht, ist ein Zugriff auf die Datei nicht mehr möglich. Falls Sie versehentlich ein Diskettenverzeichnis gelöscht haben, erhalten Sie die Fehlermeldung ERROR-170 (Datei existiert nicht) auf dem Bildschirm. Ist die Anzahl der Sektoren im Diskettenverzeichnis geringer als die tatsächliche Sektoren-Anzahl in der Datei, wird die Fehlermeldung ERROR-164 (Dateinummer nicht übereinstimmend) ausgegeben. Im letzteren Fall können Sie den Anteil der Datei, der im Bereich der angegebenen Sektoren liegt, über das nachfolgend gezeigte GetByte-Programm noch einlesen. Im Beispiel entspricht Datei 1 der zerstörten Datei, Datei 2 ist der rekonstruierbare Anteil der zerstörten Datei.

```
10 OPEN #1,4,0,"D:DATEI.1"
20 OPEN #2,8,0,"D:DATEI.2"
25 TRAP 50
30 GET #1,A
40 PUT #2,B
45 GOTO 30
50 CLOSE #1
60 CLOSE #2
```

Anmerkung: Sie können nur die Sektoren rekonstruieren, die sich vor den zerstörten Sektoren befanden. Alle hinter dem unlesbaren Bereich liegenden Sektoren sind Ihrem Zugriff entzogen. Demnach wäre es sinnvoll, die lesbaren Dateien von der defekten Diskette herunterzukopieren, um weitere Probleme zu vermeiden.

Ist die gesamte Datei zerstört, können Sie über das oben gezeigte Beispielprogramm ebenfalls alle noch lesbaren Sektoren an eine separate Datei übergeben.

Die Datei "AUTORUN.SYS"

Wenn sich auf der in Laufwerk 1 eingelegten Diskette eine Datei mit dem Namen AUTORUN.SYS befindet, wird diese bei jedem Systemstart automatisch ins RAM geladen und (sofern erforderlich) ausgeführt. Dieser Prozess wird abgeschlossen, noch bevor Sie Zugriff auf das System erhalten. Die Datei AUTORUN-.SYS kann zum einen Daten,und zum anderen nicht ausführbare oder nach Abschluß des Ladevorganges auszuführende Object Codes enthalten.

Im nächsten Beispielprogramm sehen Sie, wie über AUTORUN.SYS automatisch in die DOS-Ebene gebootet wird, selbst wenn Sie über ein fest installiertes BASIC verfügen oder eine Cartridge im Modulschacht eingesteckt ist.

Direkt nach der Programmausführung bringt Sie AUTORUN.SYS direkt zurück zur DOS-Initialisierungsroutine. Sollte dies einmal nicht der Fall sein, oder haben Sie vor dem Rücksprung die (RESET)-Taste gedrückt, müssen Sie die Initialisierung manuell abschließen, bevor Sie mit einer Programmausführung beginnen können. Hierzu können Sie zwei Speicherbereiche des Betriebssystemes, nämlich COLDST (Adresse 244 hex) und BOOT (Adresse 9 hex) modifizieren. COLDST sollte auf Null, und BOOT auf 01 gesetzt sein.

Im Beispielprogramm werden diese beiden Werte korrekt eingestellt. Die Programmausführung springt dann direkt zum Start-Vektor des DOS.

Verfügen Sie nicht über eine Assembler Editor Cartridge, können Sie unter Verwendung von BASIC POKE eine äquivalente Datei erstellen und diese Binärdatei dann im DOS abspeichern. Geben Sie diese Programmzeilen ein und starten Sie den Programmlauf:

```
10 FOR J=0 TO 10
20 READ CODE
30 POKE 15000+J,CODE
40 NEXT J
50 DATA 162,0,142,68,2,232,134,9
60 DATA 108,10,0
```

Nach Beendigung des Programmlaufes können Sie diese Datei innerhalb von DOS über Option K, BINARY SAVE, ablegen.

Beispiel:

SAVE-GIVE FILE,START,END (,INIT,RUN)
AUTORUN.SYS,3A98,3AA2,3A98

Anmerkung: Für den Parameter INIT wird kein Wert eingegeben. Wenn Sie den Computer nun kurz aus- und dann wieder einschalten, sollten Sie direkt in die DOS-Ebene booten. Um von der DOS-Ebene in BASIC zu gelangen, benutzen Sie die Funktion RUN CARTRIDGE oder drücken Sie (RESET).

Hier sehen Sie eine Assembler-Version dieses Programmes:

Hex Code	**Assembler-Sprache**
	;AUTORUN Programmlauf
	;
	;Starten von DOS, ohne Cartridge anzusprechen
	;
	COLDST = $244
	BOOT = $09
	DOSVEC = $0A
	* = 3A98
A2 00	DOSGO LDX #0
8E 44 02	STX COLDST
E8	INX
86 09	STX BOOT
6C 0A00	JMP (DOSVEC)
	* = $2E0
	;Adresse bei 2E0 starten
98 3A	.WORD DOSGO
	.END

ANHANG A: Alphabetisches Verzeichnis der reservierten Wörter in BASIC in Verbindung mit Disketten-Operationen

Nachfolgend finden Sie eine Aufstellung aller reservierten Wörter in BASIC und deren Abkürzungen, sowie eine kurze Beschreibung der daraus entwickelten BASIC-Anweisungen.

Anmerkung: Bei Verwendung der Befehls-Kurzformen muß ein Punkt nach dem Befehlswort gesetzt werden.

CLOSE	CL.	I/O-Anweisung zum Schließen einer Disketten-Datei nach Beendigung einer I/O-Operation.
DOS	DO.	Ruft das DOS-Menü auf. Das Menü enthält alle Hilfs-Optionen, die innerhalb von DOS zur Verfügung stehen. Übergibt die Programmkontrolle von der Cartridge an die DOS-Utilities.
END		Hält die Programmausführung an, schließt Dateien und schaltet SOUND-Anweisungen ab. Der Programmlauf kann mit CONT wieder aufgenommen werden. (Anmerkung: END kann in einem Programm mehrfach verwendet werden.)
ENTER	E.	I/O-Befehl zum Aufrufen eines in nicht komprimierter (untokenized) Form abgelegten Programmes. Werden mit ENTER Programme oder Programmzeilen geladen, während sich ein Programm im RAM befindet, werden diese beiden Teile miteinander verbunden. Wollen Sie dieProgramme nicht miteinander verbinden, müssenSie den Befehl NEW eingeben, bevor Sie ein Programm mit ENTER ins RAM laden.
GET	GE.	Wird bei Disketten-Operationen verwendet, um ein einzelnes Daten-Byte von einem vorgegebenen Gerät in eine spezifische Variable einzulesen.

INPUT	I.	Dieser Befehl ruft Daten von einem angegebenen Gerät ab. Das vorbestimmte Gerät ist immer E: (Bildschirm-Editor).
LIST	L.	Gibt die nicht komprimierte Form eines Programmes an ein definiertes Gerät aus.
LOAD	LO.	I/O-Befehl zum Einlesen eines in komprimierter (tokenized) Form gespeicherten Programmes von einem angegebenen Gerät.
NOTE	NO.	Speichert die absolute Sektoren-Zahl der Diskette sowie die aktuelle Byte-Zahl des Datei-Zeigers in zwei arithmetischen Variablen.
OPEN	O.	Öffnet die angegebene Datei für eine Eingabe bzw. Ausgabe. Legt die Art der Operationen fest, die für eine Datei möglich sind.
POINT	P.	Setzt den Datei-Zeiger auf eine angegebene Speicherstelle (Sektor und Byte) auf der Diskette.
PRINT	PR.	I/O-Befehl, über den eine Ausgabe vom Computer oder ? an ein angegebenes Gerät im Record-Format veranlaßt wird.
PUT	PU.	Bewirkt die Ausgabe eines einzelnen Daten-Byte (z.B. eines Zeichens) vom Computer an ein spezifiziertes Gerät.
RUN	R.	Lädt ein über die Dateiangabe spezifiziertes Programm und startet den Programmlauf.
SAVE	S.	I/O-Anweisung zum Ablegen der komprimierten Form eines Programmes in einer bestimmten Datei auf einem bestimmten Gerät.
STATUS	ST.	Ruft die STATUS-Routine für ein angegebenes Gerät auf.
TRAP	T.	Übergibt die Programmkontrolle an eine spezifizierte Zeilennummer, sobald ein Programmfehler auftritt, und ermöglicht Ihnen, Fehler zu beheben und die Programmkontrolle zu übernehmen.
XIO	X.	Allgemeine I/O-Anweisung in einem Programm, über die Sie DOS Menü-Optionen und bestimmte I/O-Befehle entwikkeln können.

Anhang B: Notationen und Terminologie in DOS 2.5

SYSTEM RESET	Drücken Sie die (RESET)-Taste auf der Tastatur
(RETURN)	Drücken Sie die (RETURN)-Taste auf der Tastatur
[ ]	Mit eckigen Klammern werden optional anzugebende Parameter gekennzeichnet
...	Wiederholungszeichen. Folgen einem in eckige Klammern gesetzten Parameter Wiederholungszeichen, kann dieser optionale Parameter beliebig oft angegeben werden.
( )	Geschweifte Klammern kennzeichnen eine Wahlmöglichkeit zwischen den einzelnen, in Klammern gesetzten Parametern. Sie dürfen jedoch nur einen einzigen der angegebenen Parameter in Ihrer Anweisung bzw. Ihrem Befehl angeben.
Großbuchstaben	Durch die Schreibweise eines Wortes in Großbuchstaben werden Befehle, Anweisungen oder andere Funktionen angegeben, die Sie über die Tastatur exakt in der vorgegebenen Form eingeben sollen.
,./:;"	Interpunktionszeichen. Diese Zeichen müssen genau in der Form eingegeben werden, in der sie innerhalb eines Befehles oder einer Anweisung erscheinen. Eckige oder geschweifte Klammern dürfen natürlich von Ihnen nicht eingetippt werden.
Befehlnr.	Befehl-Nummer. Wird innerhalb von XIO-Befehlen verwendet.
Ausdruck	In diesem Handbuch werden drei unterschiedliche Arten von Ausdrücken angegeben: Arithmetische, logische und String-Ausdrücke.
Arith.Ausdruck	Arithmetischer Ausdruck. Besteht generell aus einer Variable, Funktion, Konstante oder aus zwei durch einen arithmetischen Operator (aop) voneinander getrennten arithmetischen Ausdrücken.

Arith.Ausdruck1	Dieser arithmetische Ausdruck entspricht dem ersten Hilfssteuerungs-Byte für I/O, wenn er in Befehlen wie beispielsweise OPEN eingesetzt wird.
Arith.Ausdruck2	Dieser Ausdruck stellt das zweite Hilfssteuerungs-Byte für I/O dar, wenn er in Befehlen wie z.B. OPEN benutzt wird. Der zweite arithmetische Ausdruck ist in der Regel auf Null gesetzt.
Dateiangabe	Normalerweise ein String-Ausdruck, der sich auf eine Datei und das Gerät, auf dem die Datei abgelegt ist, bezieht (z.B. "D1:TEST.BAS" für eine Datei TEST in Laufwerk 1).
IOCB	Eingabe/Ausgabe Steuerblock (Input/Output Control Block). Ein arithmetischer Ausdruck, der einer Zahl im Bereich von 1 bis 7 entsprechen muß. Der IOCB wird als Bezug auf Geräte oder Dateien angegeben. IOCB #0 wird innerhalb von BASIC für den Bildschirm-Editor reserviert und sollte nur dann benutzt werden, wenn der Bildschirm-Editor nicht benötigt wird.
Zeilennr.	Eine Konstante, über die eine bestimmte Programmzeile in einem deferred-mode BASIC-Programm angegeben wird.
Var.	Variable eines beliebigen Typs. In diesem Handbuch wird nach arithmetischen Variablen (Arith.Var.), Matrix-Variablen (Mat.Var.) und String-Variablen (String Var.) unterschieden.
Arith.Var.	Eine Speicherstelle, an der ein numerischer Wert abgelegt ist. Variablennamen können aus mindestens einem und maximal 120 alphanumerischen Zeichen bestehen, müssen jedoch mit einem nicht umgekehrten, alphabetischen Großbuchstaben beginnen.
String Var.	String-Variable. Eine Speicherstelle, an der Zeichenketten abgelegt werden können.

Anhang C
Fehlermeldungen und Möglichkeiten zur Behebung

Anmerkung: Die Fehlermeldungen 2 bis 21 dürften eigentlich nur während der Ausführung eines BASIC-Programmes auftreten.

Fehler-Nr.	Fehlermeldung	Ursache und Behebung
2	Insufficient Memory (Unzureichende Speicherkapazität)	Ihr Computer-System hat nicht mehr genügend Speicherplatz, um eine Anweisung abzuspeichern bzw. um eine neue String-Variable zu dimensionieren. Löschen Sie alle nicht benötigten Variablennamen oder erweitern Sie Ihren verfügbaren Arbeitsspeicher (lesen Sie in Ihrem BASIC Referenzhandbuch nach, um Hinweise bezüglich Speichermanipulationen zu erhalten).
3	Value Error (Fehlerhafte Werteingabe)	Entweder war eine Integerzahl, die positiv sein müßte, negativ, oder der eingegebene Wert liegt außerhalb des als gültig definierten Bereiches. Korrigieren Sie den eingegebenen Wert.
4	Too many Variables (Zu viele Variable)	Sie haben die maximale Anzahl (128) möglicher Variablennamen überschritten. Löschen Sie alle nicht mehr benötigten Variablennamen (lesen Sie hierzu auch in Ihrem BASIC Referenzhandbuch nach).
5	String Lenght Error (Fehlerhafte String-Länge)	Sie wollten Daten aus einer Speicherstelle auslesen bzw. darin ablegen, die außerhalb der dimensionierten String-Größe liegt.

		Oder Sie haben den Wert 0 als Referenz-Index angegeben. Erweitern Sie den DIM-Bereich, bzw. verwenden Sie einen anderen Wert als Null für einen Index.
6	Out of Data Error (unzureichende Datenvorgabe)	Sie haben weniger Daten in Ihrer DATA-Anweisung angegeben, als in der darauffolgenden READ-Anweisung abgefragt werden. Geben Sie die noch fehlenden Datenwerte ein.
7	Line Number Greater Than 32767 (Zeilennummer größer 32767)	Angegebene Zeilennummer liegt außerhalb des gültigen Bereiches. Überprüfen Sie Ihre Zeilen-Referenzen in den GOTO- und RESTORE-Anweisungen etc.
8	Input Statement Error (Fehlerhafte Input-Anweisung)	Sie haben versucht, einen nicht numerischen Wert in eine numerische Variable einzulesen. Überprüfen Sie die von Ihnen verwendeten Variablen-Typen und/oder Eingabe-Daten.
9	Array or String DIM Error (Fehlerhafte Bereich- oder String-Dimensionierung)	Die DIM-Größe überschreitet den Wert 5460 für numerische Arrays bzw. 32767 für Strings. Oder: Ein Array oder String wurde redimensioniert. Oder: Sie beziehen sich auf ein nicht dimensioniertes Array/String. Überprüfen Sie das angesprochene Array hinsichtlich seiner Dimensionierung.

11	Floating Point Overflow/Underflow (Fließkomma-Überlauf/Unterlauf)	Sie haben versucht, einen Wert durch Null zu teilen, bzw. Sie beziehen sich auf eine Zahl mit einem absoluten Wert kleiner als 1E-99 oder größer gleich 1E+98. Korrigieren Sie den fehlerhaft eingegebenen Wert.
12	Line Not Found (Zeile existiert nicht)	Eine GOSUB-, GOTO- oder THEN-Anweisung bezieht sich auf eine im Programm nicht vorhandene Zeilennummer. Ändern Sie Ihre Anweisung entsprechend ab oder fügen Sie die fehlende Zeile wieder ein.
13	No Matching FOR (NEXT ohne FOR)	Eine NEXT-Anweisung wird aufgerufen, bevor die entsprechende FOR-Anweisung ausgeführt wurde. Verändern Sie die Programmstruktur, um die Programmausführung in der korrekten Reihenfolge ablaufen zu lassen.
14	Line Too Long Error (Zeile zu lang)	Sie haben die in BASIC verfügbare Zeilenlänge überschritten. Unterteilen Sie die Programmzeile.
15	GOSUB or FOR Line Deleted (GOSUB- oder FOR-Anweisung gelöscht)	Die Programmausführung trifft auf eine NEXT- oder RETURN-Anweisung, deren dazugehörige FOR- oder GOSUB-Anweisung seit dem letzten Programmlauf gelöscht wurde. Löschen Sie die NEXT- oder RETURN-Anweisung, bzw. fügen Sie den gelöschten Programmteil wieder ein.

16	RETURN Error (RETURN ohne GOSUB)	In Ihrem Programm fehlt eine GOSUB-Anweisung, auf die sich die RETURN-Anweisung beziehen kann. Ergänzen Sie Ihr Programm oder löschen Sie die RETURN-Anweisung.
17	Syntax Error (Syntax-Fehler)	Der Computer hat eine Programmzeile mit unpassender Syntax gefunden. Korrigieren Sie den Fehler.
18	VAL Function Error (Fehlerhafte VAL-Funktion)	Die Zeichenkette in einer VAL-Anweisung ist kein Zahlen-String. Korrigieren Sie Ihre Eingabe.
19	LOAD Program too Long (Zu ladendes Programm zu lang)	Ihr Computer-System kann nicht genügend Speicherplatz zur Verfügung stellen, um das aufgerufene Programm zu laden. Versuchen Sie, derzeit nicht benötigte Dateien oder Programme aus dem Arbeitsspeicher zu löschen, oder erweitern Sie den zur Verfügung stehenden Speicher.
20	Device Number Error (Fehlerhafte Geräte-Nummer)	Sie haben eine Geräte-Nummer angegeben, die außerhalb des gültigen Bereiches von 1 bis 7 liegt.
21	LOAD File Error (Lade-Fehler)	Sie haben versucht, anstelle einer komprimierten BASIC-Datei eine Datei einzulesen, die nicht geladen werden kann. Komprimierte Dateien werden mit dem Befehl SAVE erstellt.

Anmerkung: Die nachfolgenden Fehlermeldungen gelten für Eingabe-/Ausgabe-Operationen und treten in der Regel beim Ansprechen von Diskettenlaufwerken, Druckern oder anderen angeschlossenen Geräten auf. Zusätzliche Informationen zu diesen Fehlermeldungen können Sie in vielen Fällen im Bedienungshandbuch des betreffenden Gerätes finden.

128	BREAK Abort (Abbruch)	Sie haben die (BREAK)-Taste während einer I/O-Operation gedrückt und damit die Programmausführung abgebrochen.
129	IOCB* Already Open IOCB bereits geöffnet)	Eine OPEN-Anweisung innerhalb einer Programmschleife bzw. ein IOCB wird bereits für eine weitere Datei bzw. ein anderes Gerät benutzt.
130	Nonexistent Device (Gerät existiert nicht)	Sie haben versucht, ein im Geräte-Treiber nicht angegebenes Gerät anzusprechen, d.h. für das Computer-System ist das Gerät nicht existent. Diese Fehlermeldung kann dann erscheinen, wenn Sie versehentlich versuchen, beispielsweise auf ein direkt angeschlossenes ATARI-Modem Zugriff zu nehmen, ohne zuvor die "T:" Device AUTORUN.SYS - Datei gestartet zu haben. Eine weitere, häufig zutreffende Ursache für diese Fehlermeldung ist, wenn Sie vergessen, bei der Dateiangabe die Geräte-Spezifikation einzugeben (z.B. "DATEI" anstelle von "D:DATEI"). Prüfen Sie die Syntax in Ihren I/O-Befehlen nach, ob Sie das korrekte Gerät genannt haben. Dann laden und initialisieren Sie den dazugehörigen Geräte-Treiber.

131	IOCB Write-Only (IOCB gilt für Write-Only Datei)	Sie haben versucht, Daten aus einer Datei auszulesen, die für reine Schreib-Vorgänge geöffnet wurde. Öffnen Sie die Datei für Lese- und Schreibvorgänge (read and write) noch einmal.

* IOCB steht für Eingabe/Ausgabe Steuerblock

138	Device Timeout (Antwortfrist abgelaufen)	Sie haben einen Befehl über den seriellen Bus versandt, und das angesprochene Gerät hat nicht innerhalb der vom Betriebssystem vorgegebenen Frist zurückgemeldet. Ein Grund hierfür kann sein, daß Sie eine falsche Geräte-Nummer oder ein unpassendes Gerät angesprochen haben. Das Gerät ist vielleicht nicht angeschlossen, oder es ist unmöglich, innerhalb der vorgegebenen Frist eine Rückmeldung vorzunehmen. Falls es sich bei dem angesprochenen Gerät um einen Daten-Recorder handelt, könnte die Baud-Rate des Bandes falsch eingestellt sein oder das Band nicht korrekt eingelegt worden sein. Überprüfen Sie zuerst alle Anschlüsse, kontrollieren Sie, ob das Gerät eingeschaltet und auf die richtige Laufwerk-Nummer eingestellt ist (bei Diskettenlaufwerken). Vergleichen Sie, ob in Ihrem Befehl ebenfalls die richtige Laufwerk-Kennziffer angegeben wurde. Lassen Sie den Befehl dann nochmals ausführen. Tritt der Fehler ein zweites Mal auf, sollten Sie das Diskettenlaufwerk überprüfen lassen.

139	Device NAK (Ungültige Geräte-Angabe)	Das angesprochene Gerät kann aufgrund fehlerhafter Parameter (wie beispielsweise nicht adressierbarer Sektoren) keine Rückmeldung vornehmen. Das Gerät könnte in einem derartigen Fall auch unlesbare und ungültige Befehle vom Computer übermittelt bekommen haben. Prüfen Sie den verwendeten I/O-Befehl auf fehlerhafte Parameter und lassen Sie den Befehl dann nochmals ausführen. Kontrollieren Sie außerdem Ihre I/O- Kabelverbindungen. Da es sich bei Device NAK um einem Geräte-spezifischen Fehler handelt, sollten Sie im Bedienungshandbuch des entsprechenden Gerätes nachlesen.
140	Serial Frame Error (Serieller Begrenzungsfehler)	Bit 7 von SKSTAT im POKEY Chip wurde gesetzt. Das bedeutet, daß die Kommunikation zwischen dem Computer und einem Gerät gestört ist. Hierbei handelt es sich um einen relativ seltenen, aber gravierenden Fehler. Sollte diese Fehlermeldung mehr als einmal auf dem Bildschirm ausgegeben werden, sollten Sie Ihr Computer-System bzw. das angeschlossene Gerät überprüfen lassen. Sie können auch versuchen, den Defekt zu lokalisieren, indem Sie nacheinander alle angeschlossenen Geräte vom System abkoppeln. Bei einem Defekt Ihres Daten-Recorders können Sie versuchen, den Fehler gemäß den bei Fehler Nummer 138 erwähnten Möglichkeiten zu beheben.

141	Cursor Out Of Range (Cursor außerhalb des gültigen Bereiches)	Ihr Cursor liegt außerhalb des für den aktuellen Grafik-Modus gültigen Bereiches. Ändern Sie die Parameter für die Cursor-Positionierung ab.
142	Serial Bus Overrun (Datenüberlauf im seriellen Bus)	Bit 5 von SKSTAT in POKEY ist gesetzt. D.h., der Computer konnte nicht schnell genug auf einen Interrupt des seriellen Bus reagieren, oder POKEY erhielt über den seriellen Bus ein zweites 8-Bit Wort, bevor noch das zuvor übermittelte erste Wort vom Computer verarbeitet werden konnte. Dieser Fehler tritt nur sehr selten auf. Sollte die Fehlermeldung mehr als einmal ausgegeben werden, sollten Sie Ihr Computer-System zur Überprüfung geben.
143	Checksum Error (Falsche Prüfsumme)	Die Kommunikation mit dem seriellen Bus ist gestört. Die vom Gerät übermittelte Prüfsumme stimmt nicht mit dem vom Computer errechneten Wert überein. Hierfür gibt es keine standardisierte Möglichkeit zur Fehlerbehebung, da es sich sowohl um ein Hardware-, als auch um ein Software-Problem handeln könnte.

144	Device Done Error (Unpassende Geräte-Ausgabe)	Das Gerät ist nicht in der Lage, einen gültigen Befehl auszuführen. Sie haben entweder versucht, auf eine schreibgeschützte Diskette bzw. ein Gerät zu schreiben, oder das Laufwerk kann den angegebenen Sektor nicht auslesen/schreiben. Entfernen Sie ggf. den Schreibschutz der Diskette. Dieser Fehler kann auch dann auftreten, wenn Sie eine in enhanced density formatierte Diskette mit einem 810er Laufwerk lesen bzw. beschreiben wollen, oder wenn das Laufwerk eine abweichende Laufgeschwindigkeit aufweist. Sollte der Laufwerk-Motor die Ursache für den Fehler sein, müssen Sie die Diskettenstation prüfen. Lesen Sie hierzu im Bedienungshandbuch Ihres Diskettenlaufwerkes nach.
145	Illegal Screen Mode (Ungültiger Bildschirm-Modus)	Sie haben versucht, den Bildschirm-Editor über einen ungültigen Grafik-Modus zu öffnen. Kontrollieren Sie die Angaben in Ihrem Grafikmodus-Aufruf oder das aux2 Byte im IOCB.
146	Function Not Implemented (Funktion nicht implementiert)	Der Geräte-Treiber enthält die aufgerufene Funktion nicht (z.B. kann keine PUT-Anweisung oder ein bestimmter Befehl an die Tastatur gegeben werden. Überprüfen Sie den I/O-Befehl auf Verwendung des korrekten Befehles für das richtige Gerät.

147	Insufficient RAM (Unzureichender RAM-Speicher-plat)z	Für den von Ihnen gewählten Grafik-Modus steht Ihnen nicht genügend Speicherplatz im RAM zur Verfügung. Erweitern Sie Ihren Arbeitsspeicher oder verwenden Sie einen Grafik-Modus, der weniger Speicherplatz belegt.
160	Drive Number Error (Falsche Laufwerk-Kennziffer)	Sie haben für ein angesprochenes Diskettenlaufwerk eine Kennziffer angegeben, die außerhalb des gültigen Bereiches (1-8) liegt. Oder: Sie haben dem Laufwerk keinen Puffer zugewiesen. Oder: Das Laufwerk war beim Booten des Systems nicht eingeschaltet. Lesen Sie in den Abschnitten 1 und 2 dieses Handbuches nach. Prüfen Sie Ihre Angaben für die Dateispezifikation oder Byte 1802 für die Anzahl der angemeldeten Laufwerk-Puffer.
161	Too Many Open Files (Zu viele Dateien geöffnet)	Sie haben keine freien Sektor-Puffer für die Verwendung mit einer anderen Datei mehr verfügbar. Überprüfen Sie die Speicherstelle 1801 auf die Anzahl zugewiesener Sektoren-Puffer. Vergewissern Sie sich auch, daß sämtliche derzeit nicht benötigten Dateien geschlossen sind.
162	Disk Full (Diskette voll)	Auf Ihrer Diskette befinden sich keine freien Sektoren zum Speichern von Daten mehr. Legen Sie eine andere Diskette mit ausreichender Speicherkapazität ein.

163	Unrecoverable System I/O Error	Diese Fehlermeldung zeigt an, daß der Datei-Manager fehlerhaft (Nicht arbeitet. Ihre DOS-Version oder die lokalisierbarer Diskette könnte defekt sein. System I/O-Fehler) Verwenden Sie eine andere Diskette, auf der sich DOS 2.5 befindet.
164	File Number Mismatch (Unpassende Datei-Nummer)	Die Datei-Struktur wurde zerstört, oder aber die Werte für POINT wurden falsch angegeben. Eine der beiden Datei-Verbindungen zeigt auf einen Sektor, der einer anderen Datei zugewiesen ist. Schalten Sie das System ab und starten Sie den Programmlauf neu. Sollte der zweite Versuch fehlschlagen, ist Ihre Datei unwiderbringlich zerstört. Versuchen Sie, andere ggf. noch auf der Diskette befindlichen Dateien auf eine andere Diskette herunterzukopieren und formatieren Sie dann die zerstörte Diskette neu.
165	File Name Error (Fehlerhafte Dateiangabe)	Ihre Dateiangabe enthält ungültige Zeichen. Korrigieren Sie die Dateiangabe.
166	POINT Data Length Error (Fehlerhafte Daten in POINT)	Der Byte-Zähler im POINT-Aufruf war größer als 125 (bei single density Disketten). Kontrollieren Sie die Parameter in Ihrer POINT-Anweisung.
167	File Locked (Datei gesperrt)	Sie haben versucht, eine gesperrte Datei für einen anderen Zweck als für Lesevorgänge zu öffnen. Entsperren Sie die Datei mit Hilfe der Option G im DOS Menü und lassen Sie den Befehl erneut ausführen.

168	Device Command Invalid (Geräte-Befehl ungültig)	Sie haben einen für die Software-Schnittstelle des Gerätes ungültigen Befehl verwendet. Lesen Sie in der Bedienungsanleitung des betreffenden Gerätes nach und ändern Sie den Befehl entsprechend ab.
169	Directory Full (Verzeichnis voll)	Sie haben den gesamten zugewiesenen Speicherbereich für ein Verzeichnis (64 Dateien) beansprucht.
170	File Not Found (Datei existiert nicht)	Sie haben versucht, auf eine Datei zuzugreifen, die im Diskettenverzeichnis nicht aufgeführt ist. Prüfen Sie über die Menü-Option A in DOS die korrekte Schreibweise des Dateinamens und vergewissern Sie sich, daß Sie die richtige Diskette in das Laufwerk eingelegt haben.
171	POINT Invalid (Ungültige POINT-Anweisung)	Sie wollten mit POINT auf ein Byte in einer Datei zugreifen, die nicht für UPDATE-Anweisungen geöffnet war. Ändern Sie die Parameter Ihrer OPEN-Anweisung oder das aux2 Byte des IOCB, der zum Öffnen der Datei benutzt wurde.
172	Illegal Append (Ungültige Append-Funktion)	Es sollte eine DOS1 Datei für eine Append-Funktion unter Verwendung von DOS 2.5 geöffnet werden. Mit DOS 2.5 können keine Append-Funktionen auf DOS1 - Dateien angewendet werden. Kopieren Sie die Datei DOS1 mit Hilfe von DOS 2.5 auf eine DOS 2.5 Diskette.

173	Bad Sectors At Format Time (Zerstörte Sektoren beim Formatieren)	Beim Formatieren einer Diskette wurden zerstörte Sektoren aufgefunden. Da eine Diskette mit unbrauchbaren Sektoren nicht formatiert werden kann, müssen Sie eine neue Diskette verwenden. Sollte der Fehler auch bei anderen Disketten auftreten, könnte Ihr Laufwerk reparaturbedürftig sein.

Anhang D
DOS 2.5 Speicherbelegung für Computer-Systeme mit 64 K RAM

Adresse Dezimal	Hexadezimal		Speicherbelegung
65535	FFFF		Betriebssystem
49152	C000		
49151	BFFF	BFFF	
	oder oder	Keine	16K, 8K o. Keine Cartridge
32768	8000	A000	
32767	7FFF oder	9FFF oder BFFF	*
			Bereich für Bildschirm Anzeige
unterschiedlich			
unterschiedlich			HIMEM **
			Bereich für Anwenderprog.
	3305		Disk Utility Programme (DUP.SYS)
	1D7C		LOMEM **
(Puffer-Bereich ist für			Sektor-Puffer 4
DOS 2.5 reserviert!)			Sektor-Puffer 3
			Sektor-Puffer 2
			Sektor-Puffer 1
			Drive 8-Puffer (RAM)
			Drive 2-Puffer
6781	19CC		Drive 1-Puffer
6780	19CB		
			Datei-Manager Subsystem (DOS.SYS)
1792	0700		
	06FF		
			Unterschiedlich - wird oft von Programm.Sprachen benutzt (ATARI BASIC verwendet nicht $680-$6FF
	480		
	47F		

Betriebssystem
(incl. Cassetten-Puffer)

0 ***

* Abhängig vom verwendeten Grafik-Modus
** Je nach reservierter Laufwerk-Kennziffer und Sektor-Puffer unterschiedlich
*** Teile von PAGE 0 ($00 - $FF) können bei einigen Programmiersprachen verfügbar sein. Das Betriebssystem an sich benutzt nur die Adressen $00 - $7F

Anmerkung 1: Für die vorgegebenen Laufwerk- und Sektor-Speicherstellen kann LOMEM über die PEEK-Speicherstellen 2E7 (LOW) und 2E8 (HIGH) Hex bzw. über 743 (LOW) und 744 (HIGH) in Dezimal festgelegt werden.

Anmerkung 2: Um den freien Anwender-Programmierbereich oder HIMEM festzulegen, können Sie entweder die Anweisung BASIC FRE(0) verwenden oder die PEEK-Speicherstellen 2E5 (LOW) und 2E6 (HIGH) Hex bzw. 741 (LOW) und 742 (HIGH) Dezimal benutzen.

Anhang E
Umrechnungstabelle Hexadezimal / Dezimal

Vier HEX-Ziffern							
4		3		2		1	
HEX	DEZ	HEX	DEZ	HEX	DEZ	HEX	DEZ
0	0	0	0	0	0	0	0
1	4096	1	256	1	16	1	1
2	8192	2	512	2	32	2	2
3	12288	3	768	3	48	3	3
4	16384	4	1024	4	64	4	4
5	20480	5	1280	5	80	5	5
6	24576	6	1536	6	96	6	6
7	28672	7	1792	7	112	7	7
8	32768	8	2048	8	128	8	8
9	36864	9	2304	9	144	9	9
A	40960	A	2560	A	160	A	10
B	45056	B	2816	B	176	B	11
C	49152	C	3072	C	192	C	12
D	53248	D	3328	D	206	D	13
E	57344	E	3584	E	224	E	14
F	61440	F	3840	F	240	F	15

Mit dieser Tabelle können Sie bis zu vier Hex-Ziffern umrechnen.

Um beispielsweise die Hex-Zahl 1234 in Dezimal umzurechnen, addieren Sie die Angaben aller vier Spalten in der Tabelle. Für die Ziffer 1 entnehmen Sie den Wert der vierten Spalte, usw.

```
1234 hex =      4096
            +    512
            +     48
            +      4
            ________
Dezimal         4660
```

Ein anderes Beispiel:

```
EEDD hex =      57344
              +  3584
              +   208
         _____+    13
Dezimal         61149

AB hex =          160
         _____+    11
Dezimal           171
```

Anhang F
Möglichkeiten zur Beschleunigung bei der Datenübermittlung an ein Diskettenlaufwerk

DOS 2.5 bietet die Möglichkeit, Daten mit einem Verifizierungs-Durchlauf auf Diskette zu schreiben. Damit haben Sie die Sicherheit, daß für Sie wichtige Daten wirklich absolut korrekt abgelegt werden. Der Verfizierungsvorgang verlangsamt natürlich die Übertragungsgeschwindigkeit Ihrer Daten. Ihre DOS 2.5 Diskette wird mit der Voreinstellung ausgeliefert, daß bei allen Schreibvorgängen eine Datenüberprüfung vorgenommen wird.

Wenn Sie keinen übermäßigen Wert auf diese langsame, dafür aber doppelt überprüfte Möglichkeit der Datensicherung legen, können Sie diese Voreinstellung abändern. In derSpeicherstelle 1913 (dezimal) sind die Daten abgelegt, über die das Datei-Manager Subsystem die Informationen bezüglich Datensicherung erhält. (Die Angaben 57 hex bzw. 87 dez gelten für Datensicherung mit Verifizierung, 50 hex bzw. 80 dez besagt, daß die Daten ohne zusätzliche Überprüfung abgelegt werden sollen.)

Sie sollten allerdings bedenken, daß Sie durch Abschalten der Verify-Funktion zwar die Übertragungsgeschwindigkeit steigern, die Zuverlässigkeit der Datenübermittlung jedoch verringern. Um Veränderungen an dieser Voreinstellung vorzunehmen, müssen Sie für eine beschleunigte Übertragung den Befehl

POKE 1913,80

verwenden. Um die Verify-Funktion wieder einzuschalten, geben Sie entsprechend den Befehl

POKE 1913,87

ein. Wollen Sie die Vorgaben auf Ihrer DOS-Diskette so verändern, daß bei jedem Neustart des Computer-Systemes direkt die gewünschte Option gültig wird, schreiben Sie DOS und speichern die Voreinstellung über die Menü-Option H (Write DOS Files) direkt auf Diskette ab.

(Lesen Sie auch in Anhang L, dem Abschnitt über SETUP.COM, nach, um genauere Informationen über die Veränderungsmöglichkeiten der System-Konfiguration zu erhalten.)

Achtung: Das POKEn in Speicherstelle 1913 ohne Angabe eines anderen Wertes als 80 oder 87 könnte zu einem Verlust Ihrer Daten, der Zerstörung Ihrer Diskette, oder zu einem Systemabsturz führen.

Anhang G: Anmeldung angeschlossener Disketten-laufwerke an DOS

In diesem Abschnitt erfahren Sie, wie die RAM-Speicherstelle so abgeändert werden kann, daß die Anzahl der an Ihr ATARI Computer-System angeschlossenen Diskettenlaufwerke festgestellt werden kann (siehe auch Anhang L, SETUP-.COM). Wenn Sie mehr als zwei Diskettenlaufwerke (die vordefinierte Maximal-Anzahl) an Ihr Computer-System angeschlossen haben, müssen Sie den korrekten Code für die Anzahl angeschlossener Laufwerke in die RAM Speicherstelle 1802 (dez) POKEn. In der nachfolgenden Tabelle finden Sie die korrekten Eingaben für die verwendete Anzahl von ATARI 810 oder XF551 Laufwerken. Allerdings können maximal vier Diskettenstationen mit Ihrem Computer-System verbunden und mit DOS 2.5 verwendet werden, da die Kippschalter an den Laufwerken nur jeweils von 1 bis 4 eingestellt werden können. Beachten Sie, daß beim Binär-Code für die Laufwerke immer ein Wert 1 mit einem angeschlossenen Laufwerk korrespondiert.

Codes für die Anzahl angeschlossener Laufwerke

Angeschlossene Laufwerke	Dezimal-Code	Binär-Code für das Laufwerk
Laufwerk 1	01	00000001
Laufwerk 2	02	00000010
Laufwerk 3	04	00000100
Laufwerk 4	08	00001000
Laufwerk 1 + 2	03	00000011 (vordefiniert)
Laufwerk 1 + 2 + 3	07	00000111
Laufwerk 1 + 2 + 3 + 4	15	00001111

Um demnach DOS anzugeben, daß Sie drei Laufwerke an Ihr Computer-System angeschlossen haben, geben Sie den Befehl POKE 1802,7 ein.

Auch hier können Sie die vorgenommenen Einstellungen fest auf der Diskette speichern, um bei jedem Systemstart diese Konfiguration zu laden. Geben Sie hierzu DOS ein und wählen Sie im erscheinenden Menü die Option H (WRITE DOS FILES) aus. Damit speichern Sie die veränderte Version von DOS auf Ihrer Diskette ab.

Anmerkung für ATARI 130XE Anwender: Denken Sie daran, daß mit DOS 2.5 die RAM Disk immer als Drive 8 (D8:) angesprochen wird, wenn die 64 K Speichererweiterung des 130XE als RAM Disk eingerichtet wird. Lesen Sie hierzu auch in Anhang K sowie im Bedienungshandbuch Ihres 130XE Computers nach.

Technische Hinweise

Theoretisch könnten DOS 2.0S und DOS 2.5 maximal acht Diskettenlaufwerke und acht geöffnete Dateien gleichzeitig verwaltet werden. In der Praxis sind allerdings einige Einschränkungen zu beachten:

1. Die meisten Programme reservieren den IOCB #0 für Bildschirm und Tastatur (Gerät E:);

2. ATARI akzeptiert keine Laufwerk-Kennziffern größer als 4;

3. Die Datei DUP.SYS wird unter DOS 2.5 an einer bestimmten Speicherstelle abgelegt. Aus Kompatibilitätsgründen benutzt auch DOS 2.0S denselben Speicherbereich für DUP.SYS, was bei Aktivierung von DUP.SYS den verfügbaren Speicherplatz erheblich einschränkt.

Die praktische Auswirkung dieser Limitationen ist, daß Sie maximal fünf Laufwerke (inclusive einer eventuell installierten RAM Disk bei einem 130XE Computer) ansprechen können. Außerdem können Sie maximal sieben Dateien gleichzeitig geöffnet haben, wenn Sie Programmiersprachen wie BASIC, bzw. andere Programme, die Zugriff auf den LOMEM System-Zeiger (Speicherstelle $02E7) nehmen. Falls Sie zudem auf das DOS-Menü zugreifen wollen, sind Sie sogar auf nur vier Laufwerke (wieder inclusive RAM Disk) eingeschränkt. Da das DOS-Menü niemals mehr als zwei Dateien gleichzeitig benutzt, müssen Sie eine Zugriffsmöglichkeit auf mindestens diese beiden Dateien schaffen.

Eine weitere Einschränkung bei der Anzahl verwendbarer Laufwerke und Dateien ist, daß viele derzeit verfügbare Programme einen Wert für LOMEM (den Speicherinhalt von $0E27), sowie eine Mindestanzahl von zwei Laufwerken und drei derzeit geöffneten Dateien annehmen. Der Wert von LOMEM ist abhängig davon, wieviele Puffer Sie angelegt haben. Nachfolgend sehen Sie eine Tabelle, in der die Speicher-Zuweisungen für DOS 2.5 und DOS 2.0S gegenübergestellt werden.

	DOS 2.0S	DOS 2.5
Beginn Puffer	$1AC7	$19CC
Bytes pro Drive-Puffer	$80(128)	$90(144)
Bytes pro Datei-Puffer	$80(128)	$90(144)
LOMEM-Wert bei 3 Dateien, 2 Laufwerken	$1CFC	$1C6C
LOMEM-Wert bei 3 Dateien, 3 Laufwerken	$1D7C	$1CFC
LOMEM-Wert bei 2 Dateien, 4 Laufwerken	$1D7C	$1D0C
Ende DUP.SYS	$1D7C	$1D7C

Wie Sie aus dieser Tabelle ersehen können, ist $1CFC, welche von DOS 2.0S für LOMEM benutzt wird, sobald DOS für zwei angeschlossene Laufwerke und drei Dateien konfiguriert wird, die Steuer-Speicherstelle. DOS 2.5 benutzt denselben Wert für drei Dateien und drei Laufwerke, da davon ausgegangen wird, daß eine normale ATARI System-Konfiguration aus einem ATARI 130XE Computer mit zwei angeschlossenen Laufwerken und einer installierten RAM Disk besteht. Da auch eine RAM Disk einen eigenen Laufwerk-Puffer braucht, muß die Konfiguration entsprechend vorgenommen werden.

Anhang H
Die Verwendung von DOS 2.5 mit einem ATARI 810 Diskettenlaufwerk oder mit DOS 2.0S Dateien

Wenn Sie ein ATARI 810 Disketten-Laufwerk an Ihren Computer angeschlossen haben, bzw. wenn Sie mit Disketten-Dateien arbeiten wollen, die unter DOS 2.0S erstellt und gespeichert wurden, sind für die Verwendung von DOS 2.5 folgende Unterschiede zu berücksichtigen.

Die Verwendung von DOS 2.5 mit einem ATARI 810 Disketten laufwerk

- mit einem 810 Diskettenlaufwerk können Sie keine Disketten einlesen, die auf einem ATARI XF551 Laufwerk mit DOS 2.5 in enhanced density formatiert wurden. Beschriften Sie Ihre Disketten deshalb immer so, daß die Art der Formatierung (single bzw enhanced density) ersichtlich ist.

- Ihre DOS 2.5 Master-Diskette ist in enhanced density formatiert. Aus diesem Grund können Sie diese (oder in enhanced density angefertigte Sicherungsduplikate) dazu benutzen, um DOS 2.5 von einem ATARI 810 Diskettenlaufwerk einzulesen. Wenn Sie eine Version von DOS 2.5 anfertigen wollen, die auf einem ATARI 810 lauffähig ist, müssen Sie zuerst eine Diskette über DOS in single density formatieren. Danach kopieren Sie die Dateien DOS.SYS und DOS.DUP über die DOS-Menu Option H, WRITE DOS FILES, von Ihrer DOS 2.5 Master-Diskette auf diese Diskette. Sie können die Dateien RAMDISK-.COM, SETUP.COM, COPY32.COM und DISKFIX.COM über Menuoption 10, COPY FILE, auf Ihre single density Version von DOS kopieren (siehe auch Anhang K und L).

- Sie können eine in enhanced density formatierte Diskette nicht von einem XF551 Diskettenlaufwerk auf ein 810 Laufwerk kopieren. Beim Kopieren von single density Disketten hingegen ist die Art der angeschlossenen Laufwerke nicht von Bedeutung. Sollten Sie also sowohl ein XF551, als auch ein 810 Diskettenlaufwerk an Ihr Computer-System angeschlossen haben, wäre es sinnvoll, Ihre Disketten in single density zu formatieren, um dieses Kompatibilitätsproblem zu vermeiden.

- In der Regel können Sie Dateien auf enhanced density Disketten von einem XF551 Laufwerk auf single density Disketten in einem 810 Laufwerk kopieren. Hierbei sind jedoch folgende Einschränkungen zu beachten:

- Da auf Disketten mit single density erheblich weniger Speicherplatz zur Verfügung steht als auf Disketten mit enhanced density, müssen Sie beim Kopieren großer Dateien darauf achten, ob Ihnen noch genügend Speicherplatz zu Verfügung steht.

- Einzelne Dateien mit mehr als 707 Sektoren können nicht kopiert werden.

- Einige Datenverwaltungsprogramme wie beispielsweise Adresslisten, Datenbanken usw., erstellen Dateien, die direkt mit dem Hauptprogramm verknüpft sind. Solche Dateien können manchmal überhaupt nicht einzeln dupliziert werden. Und selbst wenn Sie in der Lage sind, eine derartige Datei einzeln zu kopieren, könnte es passieren, daß die Datei nur zusammen mit dem Hauptprogramm lauffähig ist. Sollten Sie mit einem derartigen Problem konfrontiert werden, lesen Sie bitte im Bedienungshandbuch des betreffenden Anwenderprogrammes nach, um festzustellen, ob das Kopieren verknüpfter Dateien überhaupt möglich ist.

Die Bearbeitung von DOS 2.0S Diskettendateien mit DOS 2.5

Sämtliche mit DOS 2.0S formatierten und beschriebenen Disketten sind vollständig kompatibel mit DOS 2.5. Umgekehrt sind alle über DOS 2.5 mit einfacher Dichte formatierten und beschriebenen Disketten kompatibel mit DOS 2.0S. Das bedeutet, daß zwischen einer DOS 2.0S Diskette und einer in single density formatierten DOS 2.5 Diskette keinerlei Unterschied im Bezug auf das Format, die Verzeichnisstruktur, Speicherkapazität, Dateistruktur usw. besteht.

Inkompatibilitätsprobleme zwischen DOS 2.5 und DOS 2.0S können nur in einer einzigen Situation auftreten - wenn Sie versuchen, eine Diskette mit DOS 2.0S einzulesen, die unter Verwendung von DOS 2.5 in erweiterter Dichte formatiert und beschrieben wurde.

Die im nächsten Absatz beschriebenen potentiellen Problemfälle basieren auf der Annahme, daß Sie das Betriebssystem DOS 2.0S mit einem XF551 Diskettenlaufwerk geladen haben und außerdem ein XF551 Laufwerk verwenden, um Dateien, die über DOS 2.5 in enhanced density formatiert wurden, zu bearbeiten.

In diesem Falle können Sie, mit einigen Ausnahmen, unter Verwendung von DOS 2.0S Daten auf die enhanced density Diskette schreiben bzw. auslesen.

- DOS 2.0S kann keine Sektoren verwalten, die außerhalb des Bereiches von 720 Sektoren liegen. Deshalb kann die erweiterte Speicherkapazität einer in enhanced density formatierten Diskette nicht genutzt werden.

Aus demselben Grund sind sämtliche Dateien, die über DOS 2.5 hinter dem Sektor 720 auf der Diskette abgelegt wurden, für DOS 2.0S nicht existent. Alle Dateien, die diesem Bereich abgelegt wurden, können mit DOS 2.0S weder bearbeitet, noch über das Diskettenverzeichnis angezeigt werden. Das nachstehende Beispiel repräsentiert die Auflistung eines Diskettenverzeichnisses unter DOS 2.5:

```
*DOS    SYS 037
*DUP    SYS 042
FILE1   DAT 204
FILE2   DAT 119
<FILE3  DAT>350
<FILE4  DAT>022
236 FREE SECTORS
```

Würden Sie das Verzeichnis dieser Diskette unter DOS 2.0S aufrufen, sähe das Verzeichnis folgendermaßen aus:

```
*DOS    SYS 037
*DUP    SYS 042
FILE1   DAT 204
FILE2   DAT 119
000 FREE SECTORS
```

Die außerhalb des Bereiches von 0 bis 720 liegenden Dateien, die im ersten Beispiel durch spitze Klammern gekennzeichnet wurden, werden von DOS 2.0S nicht berücksichtigt. Wie Sie aus dem Beispiel ersehen können, werden unter DOS 2.0S keine freien Sektoren mehr ausgewiesen, da die eigentlich noch zur Verfügung stehenden Sektoren außerhalb des akzeptierten Bereiches liegen. Im Beispiel wurde die Datei FILE3.DAT ebenfalls nicht mehr angezeigt, obwohl die dadurch belegten Sektoren sowohl unterhalb, als auch überhalb der genannten Grenze liegen. Damit wird deutlich, daß eine Datei bereits ignoriert wird, wenn nur ein geringfügiger Teil der Datei außerhalb des akzeptierten Bereiches abgelegt ist.

Selbstverständlich können alle Dateien, die von DOS 2.0S im Diskettenverzeichnis angezeigt werden, auch von DOS 2.0S bearbeitet werden. Eine mit DOS 2.0S weiterbearbeitete DOS 2.5 Diskette ist natürlich nach wie vor mit DOS 2.5 kompatibel.

Wichtiger Hinweis: Bearbeiten Sie auf keinen Fall Disketten, die unter DOS 2.5 in enhanced density formatiert wurden, mit einem unter DOS 2.0S laufenden "Fixierungsprogramm". Ein solches Programm könnte zu dem Schluß kommen, daß sämtliche von DOS 2.0S ignorierten Dateien nicht mehr lesbar, also defekt sind, und sie deshalb löschen.

Außerdem könnte das Programm die Informationen auf der Diskette löschen, aufgrund derer DOS 2.5 das Diskettenformat als enhanced density erkennt.

- Da die unter DOS 2.5 erstellten und außerhalb des von DOS 2.0S akzeptierten Bereiches abgelegten Dateien im Diskettenverzeichnis nicht aufgeführt werden, könnte es Ihnen passieren, daß Sie versehentlich einen Dateinamen zweimal verwenden. Wenn Sie beispielsweise die Datei FILE1.DAT auf der im obigen Beispiel angegebenen Diskette löschen und dafür eine neue Datei FILE3.DAT anlegen wollen, existiert der Dateiname FILE3.DAT zweimal. Sobald Sie nun das Verzeichnis dieser Diskette unter DOS 2.5 aufrufen, würden Sie zwei gleichlautende Dateien aufgelistet bekommen. DOS 2.5 ermöglicht Ihnen jedoch, eine dieser beiden Dateien umzubenennen (siehe Option E, RENAME FILE, Abschnitt 3).

Technische Hinweise

Obwohl eine single density Diskette 720 formatierte Sektoren aufweist, können sowohl von DOS 2.0S als auch von DOS 2.5 lediglich 719 Sektoren angesprochen werden. (Der Grund hierfür ist, daß DOS den Sektor mit der Kennziffer 0 nicht zur Verfügung stellt, da dieser einen Teil der Boot-Informationen enthält.)

Es gibt mehrere Programme, die auf diesen nicht verwendbaren Sektor von DOS zugreifen. In der Regel schreiben diese Programme spezielle Informationen (wie beispielsweise Kopierschutz-Daten) direkt in Sektor 720.

Bei dual density Disketten kann DOS 2.5 Sektorenkennziffern von 1 bis 1023 verwalten. Obwohl DOS 2.5 den Sektor 720 problemlos für eine Datei belegen könnte, wird dieser Sektor trotzdem wie ein bereits belegter Sektor behandelt (selbst wenn es sich um eine neue, formatierte Diskette handelt).

Der Grund hierfür ist, daß die Kombatibilität mit allen Programmen gewährleistet werden soll, die den Sektor mit der Kennziffer 720 für eigene Zwecke belegen.

Eine dual density Diskette ist in 40 Tracks a 26 Sektoren aufgeteilt (insgesamt 1040 Sektoren). Auch hier ist der Speicherplatz aufgrund der internen Verwaltung von DOS auf die Belegung der Sektoren 1 bis 1023 für Dateien beschränkt. (DOS arbeitet mit 10-Bit Sektor-Adressen, wobei die höchste durch 10-Bit darstellbare Zahl 1023 ist.) DOS 2.5 belegt Sektor 1024 mit dem Inhaltsverzeichnis der über DOS 2.0S nicht sichtbaren Dateien, d.h. über diesen Sektor werden die außerhalb des von DOS 2.0S akzeptierten Bereiches liegenden Dateien verwaltet.

Auf keinen Fall nimmt DOS 2.5 Zugriff auf die Sektoren mit den Kennziffern 1025 bis 1040. Damit bleibt dieser Bereich dem Anwender genau wie Sektor 720 bei DOS 2.0S verfügbar.

Anhang I
Die Struktur einer zusammengesetzten Binär-Datei

Die Struktur einer zusammengesetzten Binär-Datei unter Verwendung von Menü-Option C, COPY FILE

Byte-Wert	Dezimal-Wert	Hex-Wert	Beschreibung	Hex-Adresse
1	255	FF	Identifizierungs-Code (Teil 1)	
2	255	FF		
3	0	00	Startadresse (Teil 1)	$5000
4	80	50		
5	31	1F	Endadresse (Teil 1)	$501F
6	80	50		
.	.	.	Daten (Teil 1)	
.	.	.	32 Bytes	
.	.	.		
38	255	FF	Identifizierungs-Code (Teil 2)	
39	255	FF		
40	32	20	Startadresse (Teil 2)	$5020
41	80	50		
42	143	8F	Endadresse (Teil 2)	$508F
43	80	50		
.	.	.	Daten (Teil 2)	
.	.	.	112 Bytes	
.	.	.		

Die Struktur einer zusammengesetzten Binär-Datei unter Verwendung von Menü-Option K, BINARY SAVE

Byte-Wert	Dezimal-Wert	Hex-Wert	Beschreibung	Hex-Adresse
1	255	FF	Identifizierungs-Code	
2	255	FF		
3	00	00	Startadresse (Teil 1)	$5000
4	80	50		
5	31	1F	Endadresse (Teil 1)	$501F
6	80	50		
.	.	.	Daten (Teil 1)	
.	.	.	32 Bytes	
.	.	.		
38	32	20	Startadresse (Teil 2)	$5020
39	80	50		
40	143	8F	Endadresse (Teil 2)	$508F
41	80	50		
.	.	.	Daten (Teil 2)	
.	.	.	112 Bytes	
.	.	.		

Anhang J: Glossar

Abgrenzungszeichen: Ein Zeichen, durch das der Anfang oder das Ende eines Datenblocks markiert wird, das jedoch kein Bestandteil des Datenblockes ist. In BASIC werden beispielsweise Anführungszeichen (") zur Abgrenzung von Zeichenketten verwendet.

Adresse : Eine Speicherstelle im Arbeitsspeicher. In der Regel über eine 2-Byte Zahl in Hexadezimal- oder Dezimal-Format spezifiziert. (Maximaler Bereich zwischen 0 und FFFF Hexadezimal.)

Alphanumerisch: Die Großbuchstaben A - Z und Ziffern von 0 - 9 und/oder Zahlen-/Ziffern-Kombinationen. Keine Grafiksymbole, Interpunktionszeichen und andere Sonderzeichen.

Array: Eine ein- oder zweidimensionale Zusammenfassung von Elementen, auf die einzeln oder als ganze Liste über den Variablennamen und eine bzw. zwei Unterbezeichnungen zugegriffen werden kann. Array B, Element Nr. 10 wird demzufolge als B(10) bezeichnet. String-Arrays werden von BASIC nicht unterstützt, jedes Element einer Zeichenkette kann jedoch gelesen werden (z.B. A$(10.10)).Alle Arrays müssen vor ihrer Verwendung dimensioniert werden. Eine Matrix ist ein zweidimensionales Array.

ATASCII: Kodiermethode zum Speichern von Text-Dateien. In ATASCII (einer modifizierten Version von ASCII, A merican S tandard C ode for I nformation I nterchange) wird jedem Zeichen, jedem Grafik-Symbol und den meisten Kontrolltasten eine entsprechende Kennziffer zugewiesen. Diese Kennziffer entspricht einem 1- bzw. 2-Byte Code (Dezimal 0 bis 255). Tabelle siehe ATARI BASIC Bedienungshandbuch.

Ausgabe: Siehe "Eingabe".

AUTORUN.SYS: Vom Betriebssystem reservierter Dateiname.

Baud Rate: Übertragungsrate von Daten in Bits pro Sekunde.

Binär: Auch Basis 2 genannt. Ein Zahlensystem, bei dem lediglich die Ziffern 0 und 1 benutzt werden.

Binary Load: Laden einer binären Objekt-Datei in Maschinensprache in den Arbeitsspeichers des Computers,

Binary Save: Speichern einer binären Objekt-Datei in Maschinensprache auf ein Diskettenlaufwerk bzw. einen Programm-Recorder.

Bit: Abkürzung für "Binärziffer" ("Binary Digit"). Kleinste Einheit von Informationen, dargestellt durch den Wert 0 oder 1.

Boot: Initialisierungsprogramm, über das der Computer nach dem Einschalten seine Startinformationen erhält. Nach Abschluß des Boot- oder Start-Vorganges ist der Computer in der Lage, beispielsweise Anwenderprogramme zu laden und auszuführen.

Break: Abbruch eines Programmlaufes. Durch Drücken der (BREAK) Taste wird die Programmausführung abgebrochen.

Byte: 8 Bits. Ein Byte kann ein Zeichen repräsentieren. Ein Byte hat als gültigen Bereich die Werte 0 bis 255 (dezimal).

CIO: (" C entral I nput/ O utput Subsystem"). Der Teil des Betriebssystemes, über den Eingabe-/Ausgabe-Operationen gehandhabt werden.

CLOSE: Beenden des Zugriffs auf eine Diskettendatei. Nach dem Schließen einer Datei mit dem Befehl CLOSE können Sie auf diese Datei nicht mehr zugreifen. Zur Weiterbearbeitung wird der Befehl OPEN verwendet.

Datei: Eine organisierte Ansammlung zusammengehörender Daten. Eine Datei ist die größte Gruppierung von Information, die mit einem einzelnen Namen adressiert werden kann.

Dateizeiger: Ein Zeiger auf eine bestimmte Speicherstelle in einer Datei. Jede Datei verfügt über eigene Zeiger.

Dateiangabe: Folge von Zeichen, über die ein bestimmtes Gerät sowie ein Dateiname spezifiziert werden.

Daten: Jede Art von Information, in der Regel als Zusammenfassung von Bytes ausgedrückt.

Debug: (=Fehlerbehebung) Routine zur Aufdeckung und Behebung von Fehlern in einem Programm.

Dezimal: Auch Basis 10 genannt. Ein Zahlensystem, bei dem lediglich die Ziffern 0 bis 9 Gültigkeit haben. Dezimalzahlen werden im Computer in binär kodiertem Dezimalformat abgespeichert. Siehe auch Bit, Hexadezimal und Oktal.

Dichte: ("Density"). Ausnutzung des zur Verfügung stehenden Speicherplatzes auf einem Speichermedium, z. B. die Anzahl von Sektoren pro Spur auf einer Diskette. Beim Formatieren sowohl mit einfacher Dichte, als auch mit erweiterter Dichte, werden 128 Byte pro Sektor abgelegt.

Diskette: Auch Disk genannt. Ein Aufzeichungs-/Wiedergabe-Medium, vergleichbar mit einem Datenband, allerdings in Form einer flachen Scheibe und geschützt durch eine stabile Plastikhülle. Die Zugriffszeit auf eine Diskette ist erheblich kürzer, verglichen mit einem Datenband.

DOS: Abkürzung für das Disketten-Betriebssystem (D isk O perating S ystem). Software, über die der Einsatz eines Diskettenlaufwerkes ermöglicht wird.

DOS.SYS: Vom Betriebssystem reservierter Dateiname.

Eingabe: Datenübertragung von einem externen Gerät (wie z.B. einem Laufwerk) in das RAM des Computers. Das Gegenstück hierzu ist "Ausgabe". Die Begriffe "Eingabe" und "Ausgabe" werden oft kombiniert verwendet (Eingabe/Ausgabe, Input/Output oder einfach I/O), um Datenübertragungs-Operationen anzugeben. Bezugspunkt ist immer der Computer, d.h. eine Eingabe geht an, und eine Ausgabe kommt vom Computer.

End of File: Eine Markierung, über die dem Computer angezeigt wird, daß das Ende einer Datei auf der Diskette erreicht wurde.

Extender: An den eigentlichen Dateinamen angesetzter, aus drei Buchstaben bestehender Nachsatz. Wird von den meisten Programmen automatisch an die erstellten Dateien angefügt und kennzeichnet die Art der abgelegten Datei. Muß durch ein Punktzeichen vom Dateinamen abgegrenzt werden.

Formatieren: Dient zum Einrichten einer neuen oder neu zu beschreibenden Diskette in einzelne Spuren und Sektoren. Beim Formatieren in einfacher Dichte wird die Diskette in 40 kreisförmige Spuren, die wiederum in jeweils 18 Sektoren unterteilt sind, aufgeteilt. Bei erweiterter Dichte erfolgt die Unterteilung einer Diskette in 40 Spuren mit je 26 Sektoren. In jedem Sektor können 128 Bytes an Daten abgelegt werden.

Hexadezimal bzw. Hex: Auch Basis 16 genannt. Zahlensystem, bei dem die sechzehn alphanumerischen Zeichen 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E und F Gültigkeit haben.

Höchstwertiges Byte: Das am weitesten links bzw. oben stehende Byte in einer Zahl oder einem Wort.

I/O: Siehe unter "Eingabe".

Indizierte Adressierung: Siehe unter "Random Access".

Input: Siehe unter "Eingabe".

INPUT: Ein BASIC-Befehl, über den numerische oder String-Daten von einem bestimmten Gerät abgerufen werden.

IOCB: I nput/ O utput C ontrol B lock. Ein Teil des RAM, der für die Adressierung von I/O-Geräten, sowie für die Verarbeitung der übermittelten bzw. empfangenden Daten reserviert ist.

iocb: Ein arithmetischer Ausdruck mit einem Wert von 1 bis 7. Wird zur Bestimmung eines Gerätes bzw. einer Datei benutzt.

Kilobyte oder K: 1024 Bytes. Ein 16K RAM verwaltet demnach 16 x 1024 = 16.384 Bytes.

Laufwerk-Kennziffer: Integerzahl zwischen 1 und 8, über die das angesprochene Laufwerk spezifiziert wird.

Laufwerk-Spezifikation: Teil der Dateiangabe, über die dem Computer das anzusprechende Laufwerk mitgeteilt wird. Wird diese Angabe weggelassen, spricht der Computer automatisch Laufwerk 1 an.

Letztwertiges Byte: Das am weitesten rechts bzw. unten stehende Byte in einer Zahl oder einem Wort.

Maschinensprache: Befehlsfolge für den in einem Computer verwendeten Mikroprozessor-Chip.

Null-String: Eine Zeichenkette, die keine Zeichen enthält. Beispielsweise wird mit A$="" ein String A als Null-String abgelegt.

Objekt-Code: Vom "Quell-Code" (normalerweise von einer Assemblersprache) abgeleitete Maschinensprache.

Oktal: Auch Basis 8 genannt. Das Oktalzahlensystem arbeitet mit den Ziffern 0 bis 7. Adressen- und Bytewerte werden oft in Oktalwerten ausgedrückt.

OPEN: Vorbereitung einer Datei zur Bearbeitung. Mit OPEN wird angegeben, ob eine Ein- bzw. Ausgabe-Operation ausgeführt werden soll, zusammen mit der Angabe, welches Gerät angesprochen wird.

Output: Siehe "Eingabe".

Parameter: Variable in einem Befehl oder einer Funktion.

Peripherie-Gerät: Ein I/O-Gerät.

POKEY: Ein bekannter I/O Chip, über den die Kommunikation mit dem seriellen Bus verwaltet und Töne über den Fernseh-Lautsprecher generiert werden.

Puffer: Ein temporärer Speicherbereich im RAM. Wird zur Aufbewahrung von Daten zur weiteren Verarbeitung bzw. für Eingabe-/Ausgabe-Operationen usw. benutzt.

Quelladresse bzw. Quellgerät: (= "Source"). Gerät oder Adresse, das die Daten enthält, die an ein bestimmtes Gerät bzw. eine Adresse geleitet werden sollen. Siehe auch "Destination".

Random Access: Methode zum direkten Einlesen von Daten in bestimmte Sektoren und Bytes, ohne hierfür die gesamte Datei sequentiell durchlaufen zu müssen.

Record: Ein Datenblock, der durch EOL-Zeichen (Zeilenende, 9B Hex) abgegrenzt ist.

Schreibschutz: Methode zum Schützen von Disketten vor versehentlichem Überschreiben. Die meisten Disketten werden durch Überkleben der Aussparung in der Diskettenhülle schreibgeschützt.

Sektor: Kleinster Block von Daten, der auf Diskette geschrieben, bzw. davon ausgelesen werden kann. Eine in single density formatierte Diskette kann 128 Daten-Bytes abspeichern.

Sequentieller Zugriff: Methode zum Auslesen von Daten von einer Diskette in sequentieller Folge, beginnend beim ersten Byte der Datei.

Start Punkt: Eine Adressangabe, welche die Ausführung eines Maschinenprogrammes oder einer Routine beginnt. Auch genannt "Übertragungsadresse".

Source Code: Auch Quell-Code genannt. Eine Folge von Anweisungen, geschrieben in einer anderen Sprache als Maschinensprache. Daher ist eine Übersetzung der Anweisungen wärend des Programmlaufes notwendig.

Spur: Ein Kreis auf der Diskette, der für das magnetische Speichern von Daten verwendet wird. Jede Spur (auch "Track" genannt) enthält 18 Sektoren bei single density Disketten und 26 Sektoren bei enhanced density Disketten. Jeder Sektor kann 128 Bytes an Daten abspeichern. Auf jeder Diskette werden 40 Spuren angelegt.

String: (="Zeichenkette"). Folge von Buchstaben und/oder Zahlen. In der Regel abgegrenzt durch Anführungszeichen (").

System-Diskette: Exakte Kopie einer Original-Masterdiskette. Arbeiten Sie immer mit Backup-Kopien Ihrer Original-Programmdisketten und fertigen Sie von allen wichtigen Programm- oder Datendisketten Duplikate an.

Tokenized: Verfahren, bei dem ein BASIC Source Code interpretiert und dann in das vom BASIC Interpreter intern benutzte Format umgewandelt wird.

Variable: Adresse, an der ein bestimmter Wert abgelegt werden kann. In Variablen abgelegte Werte sind in der Regel Zahlen und/oder Zeichenketten.

Verzeichnis: Auflistung aller Dateien, die auf einer Diskette abgelegt oder gespeichert wurden. Enthält Dateinamen und Dateilängen.

Vordefinierung: Bedingungen oder Werte, die vom Computer angenommen werden, solange Sie diese nicht abändern. In BASIC wird beispielsweise der Grafik-Modus mit GRAPHICS 0 vordefiniert und bleibt solange gültig, bis Sie diese Einstellung verändern.

Zieladresse bzw. Zielgerät: Gerät oder Adresse, an das bzw. die während eines Informationsaustausches (insbesondere bei I/O-Operationen) Daten übertragen werden. Siehe auch unter Quelladresse bzw. Quellgerät:

Anhang K
Die Verwendung von DOS 2.5 mit einer ATARI 130XE RAM-Disk

Der ATARI 130XE Computer hat eine Speicherkapazität von 131.072 Bytes (d.h. 128 K) als RAM. Damit steht Ihnen der zweifache Speicherplatz im Vergleich zu älteren ATARI Computer-Systemen zur Verfügung. Die zusätzlichen 64 K RAM bringen Ihnen mehrere Vorteile: Die Animation von Bildern auf dem Bildschirm läuft erheblich schneller ab, der zur Verfügung stehende Speicherplatz kann für umfangreiche Dateien wie z. B. Datenbanken genutzt werden, usw.

Sie können den zusätzlichen Speicherbereich des 130XE Computers aber auch als extrem schnelles, faktisches Diskettenlaufwerk einsetzen. Wird dieser Bereich als RAM Disk angemeldet (von DOS 2.5 über Laufwerk-Kennung 8 angesprochen), können Sie wie auf einen erweiterten Speicherbereich von 499 Sektoren auf einer Diskette darauf zugreifen. Diese Speichermenge enspricht beinahe der Hälfte einer in enhanced density formatierten Diskette.

Selbstverständlich ist dieser zusätzliche Speicherbereich nur temporär nutzbar, d.h., Sie verlieren sämtliche darin abgelegten Daten beim Abschalten Ihres Computer-Systemes, sofern Sie diese nicht vorher auf einem externen Speichermedium wie beispielsweise einer Diskette abgespeichert haben.

Trotz dieser Einschränkung ist eine RAM-Disk ein äußerst nützliches Hilfsmittel. Sie können fast verzögerungsfrei zwischen BASIC (bzw. einer anderen Programmiersprache) und DOS hin- und herschalten. Sie können auch mit in Laufwerk 8 "gespeicherten" Dateien arbeiten, was besonders bei der Übertragung von großen Datenmengen zwischen zwei Programmen (z.B. wenn ein Programm die Ausführung eines anderen Programmes veranlaßt) nützlich ist.

Anmelden einer RAM-Disk

Auf Ihrer DOS 2.5 Master-Diskette befindet sich eine Datei mit dem Namen RAMDISK.COM, über das automatisch die zusätzlichen 64 K des 130XE Computers als RAM-Disk angemeldet werden können. (Wollen Sie keine RAM-Disk installieren, lesen Sie bitte im nächsten Abschnitt nach.)

Sobald Sie Ihr 130XE Computer-System mit einer DOS 2.5 Master-Diskette booten, auf der die Datei RAMDISK.COM enthalten ist, wird automatisch

- eine Meldung auf den Bildschirm gebracht, daß die RAM-Disk installiert wird
- der zusätzliche Speicherbereich des 130XE Computers wie ein Diskettenlaufwerk eingerichtet, auf das DOS 2.5 über die Kennung 8 zugreifen kann
- die Datei DUP.SYS sowie MEM.SAV von der DOS 2.5 Master-Diskette auf die RAM-Disk kopiert. Auf diese beiden Dateien wird zukünftig nur noch über die RAM-Disk, und nicht mehr über die DOS 2.5 Diskette Zugriff genommen.

Sie können die verwendbare Speicherkapaziät Ihrer RAM-Disk mit Hilfe der Dateien DUP.SYS und MEM.SAV erweitern. Hierzu

- verändern Sie den Inhalt von Speicherstelle 5439 ($153F) auf ATASCII1 (z.B. POKE 5439.ASC("1")
- löschen Sie die Dateien DUP.SYS und MEM.SAV aus dem RAM (Kennung 8) unter Verwendung der Menü-Option D, DELETE FILE(S) und Angabe von D8:*.* hinter der Abfrage der zu löschenden Datei.

Anmerkung: Wenn Sie Ihr System mit einer Diskette booten, auf der sich die Datei DUP.SYS nicht befindet, wird über RAMDISK.COM zwar eine RAM-Disk angemeldet, die Dateien DUP.SYS und MEM.SAV werden jedoch nicht ins RAM kopiert.

Verwendung von DOS 2.5 mit einer RAM-Disk

Wegen der Größe der RAM-Disk können Sie die DOS Menü-Option J, DUPLICATE DISK, nicht zum Kopieren von kompletten Disketten in die RAM-Disk benutzen. Stattdessen müssen Sie die gewünschten Dateien einzeln umkopieren. Achten Sie darauf, daß die Dateilänge der zu kopierenden Datei nicht den zur Verfügung stehenden Speicherbereich auf der RAM-Disk überschreitet. Sie können über DOS allerdings den Inhalt der RAM-Disk auf eine Diskette kopieren.

Auf dieser Diskette können Sie unter DOS jedoch nur noch auf 499 Sektoren mit Daten zugreifen, obwohl der gesamte Disketteninhalt jederzeit wieder vollständig auf die RAM-Disk zurückkopiert werden kann.

Die Arbeit mit einem 130XE Computer ohne angemeldete RAM-Disk

Wenn Sie keine RAM-Disk für Ihren ATARI 130XE Computer anmelden wollen, müssen Sie die Datei RAMDISK.COM auf Ihrer DOS 2.5 Master-Diskette entweder löschen oder umbenennen. Sie können dann den zusätzlichen Speicherbereich des 130XE für andere Zwecke nutzen.

Wenn Sie nur zeitweise mit Anwenderprogrammen arbeiten, bei denen die Verwendung einer RAM-Disk nicht sinnvoll ist, sollten Sie die Datei RAMDISK-.COM unverändert auf Ihrer DOS 2.5 Master-Diskette belassen und sich stattdessen eine Arbeitskopie der DOS-Diskette anfertigen, von der Sie diese Datei löschen. Oder Sie benennen die Datei RAMDISK.COM um, um jederzeit wieder Zugriff darauf nehmen zu können, wenn Sie eine RAM-Disk anmelden wollen.

Anhang L: Laden der DOS 2.5 Utilities

Die DOS 2.5 Utilities

Mit Utility bezeichnet man Hilfsprogramme zur System-Steuerung und Programmierhilfen. Auf der DOS 2.5 Master-Diskette befinden sich drei solcher Programme:

COPY23.COM setzt Dateien, die mittels DOS 3 auf Diskette geschrieben wurden, in das DOS 2.5 Format um.

DISKFIX.COM hilft Ihnen bei verschiedenen Problemen, die mit DOS 2.5 auftreten können.

SETUP.COM gestattet Ihnen die Änderung wichtiger DOS 2.5 Parameter.

Laden eine Utilities

Die Hilfsprogramme sind Binär-Dateien und werden mit Option "L" vom DOS-Menü aufgerufen. Wenn Sie beispielsweise mit COPY23.COM arbeiten möchten, geben Sie den Buchstaben "L" und anschließend den Dateinamen COPY23.COM ein.

COPY23.COM

Dieses Utility arbeitet wie die Kopier-Funktionen des DOS-Menüs, nur kopiert es zwischen zwei völlig unterschiedlichen Formaten. Nach dem Laden werden Sie aufgefordert, die Nummern der Laufwerke anzugeben, in welchen sich die DOS 3 ("VON") und die DOS 2.5 ("AUF") Disketten befinden. Bei nur einem Laufwerk geben Sie jeweils die 1 ein – Sie müssen dann die Disketten abwechselnd einlegen. COPY23 zeigt Ihnen das Inhaltsverzeichnis de DOS 3 Diskette an, aus dem Sie anhand der Nummer eine Datei zum Kopieren auswählen können. Diese Datei wird dann in den Computer geladen. Beim Betrieb mit nur einem Laufwerk müssen Sie nun die Disketten wechseln und die START-Taste drücken. Vom Speicher des Computers überträgt COPY23 die Datei auf die DOS 2.5 Diskette. Längere Dateien können evtl. mehrere Durchgänge erforderlich machen.

COPY23 kann Dateine, deren Länge 124.700 Bytes übersteigt, nur mit zwei Diskettenstationen konvertieren. Sollte während des Kopiervorganges ein Fehler auftreten, gibt COPY23 eine Fehlermeldung mit der entsprechenden Fehlernummer aus und bricht den Lauf ab.

DISKFIX.COM

DISKFIX ist ebenso wie DOS 2.5, menügesteuert. Sie können unter fünf Optionen wählen:

1) LAUFWERKS #
2) UNLOESCHEN
3) DISK PRUEFEN
4) NAME NACH NUMMER AENDERN
5) DOS

1. Laufwerks-Nummer

Mit Option "1" bestimmen Sie, mit welchem Lauferk DISKFIX arbeiten soll. Drücken Sie einfach die entsprechende Zifferntaste (1, 2, 3, 4, oder 8); Sie gelangen dann wieder ins Menü, wobei DISKFIX die neue Laufwerks-Nummer anzeigt. Eine ungültige Eingabe führt nicht zu einer Änderung.

2. "UNLOESCHEN"

Irren ist menschlich! Wenn Sie versehentlich eine Datei gelöscht haben sollten, können Sie diese mit der DISKFIX-Option "2" evtl. wieder herstellen. Voraussetzung dafür ist, daß Sie zwischdurch noch keine Schreib-Operation auf der Diskette ausgeführt haben. DOS 2.5 kennzeichnet gelöschte Dateien im Inhaltsverzeichnis und gibt die belegten Sektoren wieder zum Beschreiben frei. Die Datei ist also auch im gelöschten Zustand nich intakt. Erst, wenn die betreffenden Sektoren von einer anderen Datei wieder benutzt werden, ist die zuvor gelöschte Datei verloren.

DISKFIX kann Ihnen u.U. auch wieder zu Dateien Zugang verschaffen, die zwar einmal eröffnet, aber später nicht abgeschlossen wurden, weil Sie beispielsweise versehentlich das Programm unterbrochen hatten.

Nach Anwahl der Option "2" legen Sie bitte die Diskette mit der betreffenden Datei in die Arbeits-Station ein und drücken die START-Taste. Daraufhin erscheint auf dem Bildschirm eine Liste der ersten 32 Dateien dieser Diskette. Gelöschte Dateien erkennen Sie leicht am Buchstaben "D" vor dem Dateinamen, offene Dateien sind mit "W" gekennzeichnet. Geben Sie nun die Nummer der Datei ein, die Sie wieder bearbeiten wollen, oder drücken Sie nur RETURN – dann werden Ihnen die nächsten, bis zu 32 Dateien angezeigt. DISKFIX gibt Ihnen zu der gewählten Datei einige zusätzliche Informationen, die besonders für Programmierer wichtig sind, und fragt, ob Sie diese Datei wieder herstellen möchten. Nach Ihrer Eingabe prüft DISKFIX die auf der Diskette befindlichen Dateien (siehe dazu auch Option "3"). Anschließend ist die Datei wieder mit dem normalen Status im Inhaltsverzeichnis enthalten und kann von dort aus wieder aufgerufen werden.

3. DISK PRUEFEN

DOS 2.5 erstellt auf jeder Diskette eine Tabelle der verfügbaren, also freien Sektoren, die sich VTOC nennt (Volume Table Of Contents). Die VTOC wird bei jeder Schreib-Operation aktualisiert, sofern kein Fehler auftritt. Die DISK-FIX-Option "3" überprüft diese Tabelle anhand der gespeicherten Daten und schreibt sie neu. Offene Dateien erhalten dabei den Status "gelöscht", und die zuvor belegten Sektoren werden freigegeben.

4. NAME NACH NUMMER AEDNERN

Für den Fall, daß bei mehreren Dateien Namensgleichheit besteht (siehe DOS 2.5, Option "E"), können Sie mit der DISKFIX-Option "4" auch den Namen der zweiten Datei ändern. DISKFIX zeigt Ihnen wieder das Inhaltsverzeichnis an, aus dem Sie durch Eingabe der entsprechenden Zahl eine Datei auswählen können. Wenn Sie die Frage "DIESES FILE" mit "J" beantworten, fragt DISKFIX Sie nach dem neuen Namen. Beachten Sie bitte die Hinweise über Dateinamen unter DOS 2.5.

5. DOS

Die Option "5" bricht DISKFIX ab und führt Sie wieder in das DOS 2.5-Menü zurück.

SETUP.COM

Nach dem Laden des Utilities SETUP.COM erscheint folgendes Menü:

1. LAUFWERKS-NUMMER AENDERN
2. SYSTEM-EINSTELLUNG AENDERN
3. AUTORUN DATEI ERSTELLEN
0. ZURUECK ZUM DOS

1. Laufwerks-Nummer ändern

In dem Text über dem Menü wird Ihnen mitgeteilt, auf welche Diskettenstation die von Ihnen gewünschten Optionen wirken. Mit Option "1" können Sie ein anderes Laufwerk (1 bis 4)bestimmen. Die 130xe RAM-Disk kann nicht benutzt werden.

2. System-Einstellung ändern

Nach Anwahl dieser Option wird die derzeitige System-Einstellung auf dem Bildschirm ausgegeben:

AKTIVE LAUFW.: 1 2 8 (RAM-Disk)

MAX. 3 DATEIEN GLEICHZEITIG OFFEN

SCHREIBEN ERFOLGT MIT PRUEFUNG

WOLLEN SIE DIESE EINSTELLUNGEN
VERAENDERN (J/N)?

Wenn Sie nun "J" eingeben, fragt SETUP die neuen Einstellungen nacheinander ab.

Jede angemeldete Diskettenstation benötigt 144 Bytes und jede offene Datei 128 Bytes des RAM Speichers. Sie verlieren mit jedem angemeldeten Laufwerk einen kleinen Teil des Speichers, gewinnen jedoch viel Platz auf der Diskette. Beachten Sie bitte, daß mit vier Laufwerken und der RAM-Disk oder nur einer möglichen offenen Datei die DOS Optionen "C", "D" und "O" nicht einwandfrei arbeiten. SETUP gibt Ihnen dann eine Warnung vor möglichen Fehlern.

Beim Schreiben auf die Diskette prüft DOS 2.5, ob die Daten fehlerfrei geschrieben wurden. Übertragungsfehler treten nur sehr selten auf, jedoch kann ein einziges fehlerhaftes Byte ein Programm unbrauchbar machen! Zur Zeitersparnis bei Schreib-Operationen kann diese Prüfung ausgeschaltet werden, doch ist dis nur bei Daten-Disketten (Texte usw.) angeraten.

Nach der Eingabe der neuen Einstellungen zeigt SETUP diese an, und Sie müssen die Werte mit "J" bestätigen. Sie können dann auch entscheiden, ob eine permanente Speicherung auf die Diskette erfolgen soll.

3. AUTORUN Datei erstellen

Beim Einschalten des ATARI Computers wird das DOS 2.5 in den Speicher geladen und auf der Diskette nach einer Datei mit dem Namen AUTORUN.SYS gesucht. Ist diese Datei vorhanden, so wird sie geladen und gestartet, bevor die System-Kontrolle an BASIC übergeben wird. Mit der Option "3" können Sie eine solche Datei erstellen, die

1. die Treiber-Software für das ATARI 850 Interface-Modul initialisiert (sofern es angeschlossen ist),
2. ein BASIC-Programm lädt und startet oder
3. beides zusammen durchführt.

Wenn Sie Punkt "2" oder "3" wählen, fragt SETUP nach dem Namen Ihres BASIC-Programms und prüft nach Ihrer Eingabe, obdieses Programm auf der Diskette existiert. Danach wird eine Datei AUTORUN.SYS auf die Diskette geschrieben, und Sie gelangen wieder ins Menü.

Anhang M
Spezifikationen des ATARI XF551 Diskettenlaufwerkes

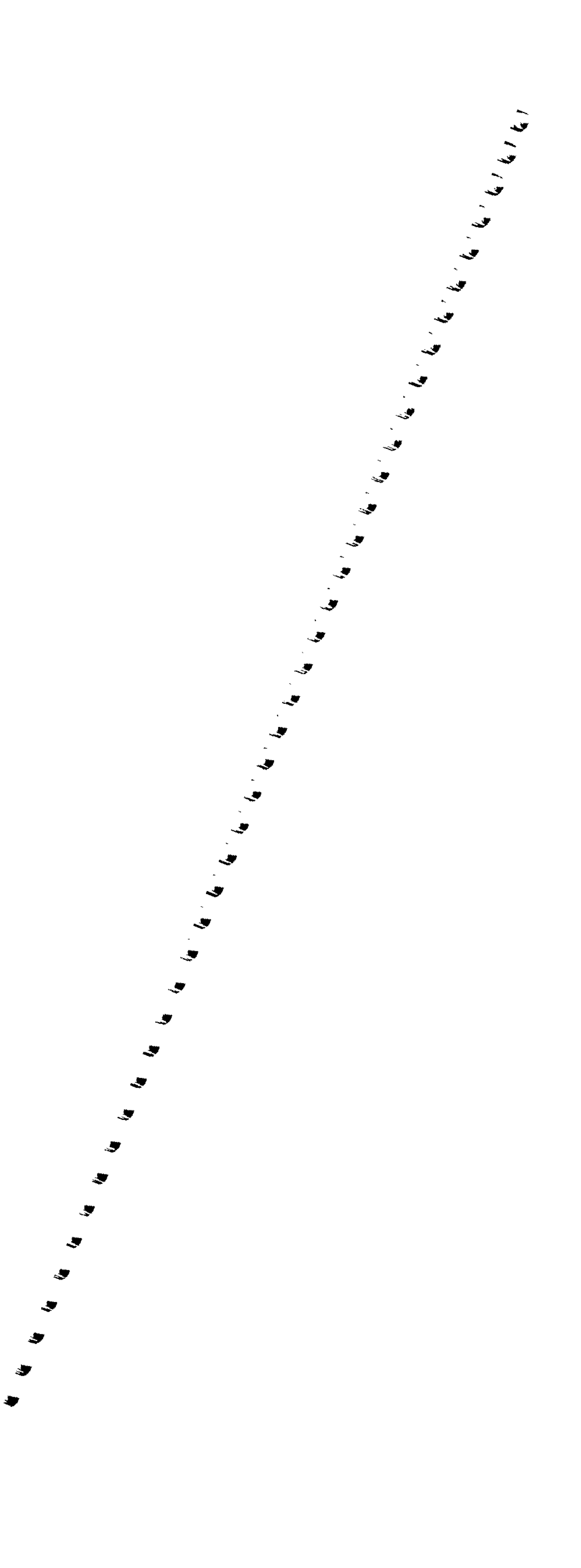
Kontrolleinheit	**6507 Mikroprozessor**
Diskettenformat	dual density, einseitig, 5 1/4 Zoll
Speicherkapazität	127 K
Abmessungen:	
- Höhe	70 mm
- Breite	180 mm
- Länge	290 mm
Gewicht	2.7 kg
Stromversorgung	8.52 Volt AC 3 Hz
Stromverbrauch:	
- eingeschaltet	15 Watt
- in Betrieb	30 Watt
- beim Einschalten	50 Watt
Temperatur:	
- in Betrieb	+5°C bis +45°C
- Lagerung	−20°C bis +65°C
Relative Luftfeuchtigkeit:	
- in Betrieb	20 bis 80 %
- Lagerung	5 bis 95 %

	dual density	single density	RAM Disk
Anzahl Spuren	40	40	./.
Anzahl Sektoren/Spur	26	18	./.
Gesamtanzahl Sektoren	1.040	720	512
Anzahl der für DOS verfügbaren Sektoren	1.023	719	511
DOS Overhead in Sektoren	13	12	12
Anzahl verfügbarer Sektoren für Datenspeicherung	1.010	707	499
Anzahl Bytes pro physikal. Sektor	128	128	128
Anzahl Bytes Overhead pro logischem Sektor	3	3	3
Anzahl der für Datenspeicherung verwendbaren Bytes pro Laufwerk	126.250	88.375	62.375
Decodierungs-Verfahren	MFM	FM	
Übertragungsgeschwindigkeit in Byte pro Minute	250.000	125.000	
Zugriffszeit Spur auf Spur	40 MS	40 MS	
Motorstart (max.)	1000 MS	1000 MS	

Kundendienst:

Sollte während der Garantiezeit ein Defekt an Ihrem Gerät auftreten, wenden Sie sich bitte an Ihren Händler, von dem Sie Ihre Atari Floppy gekauft haben.

Sollte ein Defekt nach Ablauf der Garantiezeit auftreten, wenden Sie sich bitte an Ihren Atari Händler.



ATARI

Atari Corp. (Deutschland) GmbH
Frankfurter Straße 89-91
6096 Raunheim

Printed in Germany